

Laitetilan etäseuranta

Raspberry Pi & SNMP

Joonas Holmén

Opinnäytetyö

Huhtikuu 2016

Tekniikan ja liikenteen ala

Insinööri (AMK), Tietotekniikan koulutusohjelma

Tekijä(t) Holmén, Joonas	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 4/2016
	Sivumäärä 96	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Laitetilan etäseuranta Raspberry Pi ja SNMP		
Tutkinto-ohjelma Insinööri (AMK), Tietotekniikan koulutusohjelma		
Työn ohjaaja(t) Mika Rantonen, Olli Väänänen		
Toimeksiantaja(t) Jyväskylän Ammattikorkeakoulu, Juha Jokinen		
<p>Tiivistelmä</p> <p>Opinnäytetyön toimeksiantajana toimi Jyväskylän ammattikorkeakoulu. Toimeksiantajalla oli tarve seurata laitetilan ilmastoa, keskeytymättömän virransyötön tilannetietoja sekä näiden muutoksia. Tavoitteena oli toteuttaa järjestelmä, jolla seurattavat kohteet saadaan luettavaksi käyttäen SNMP-ohjelmistoa sekä graafinen käyttöliittymä, jolla ilmaston tilaa ja muutoksia voidaan monitoroida.</p> <p>Työn alustaksi määritettiin Raspberry Pi 2 -pienetietokone. Järjestelmä rakentui avoimen lähdekoodin ohjelmistoista. Järjestelmä rakentui kahdesta osa-alueesta, etäkyselystä sekä valvonnasta. Etäkyselyllä ilmaston tila-arvoja kerätään ja asetetaan saataville SNMP:llä. Etäkyselyä varten luotiin ohjelmia, joilla kerääminen sekä olio-alipuun luominen tapahtuu. Valvonta toteutettiin Cacti-valvontaohjelmistolla, joka ylläpitää tietokantaa arvoista ja kohteista, sekä sallii käyrien piirtämisen arvojen muutoksista.</p> <p>Cactin toimintaa laajennettiin lisäämällä siihen hälytysjärjestelmä. Hälytysjärjestelmä kykeni lähettämään tekstiviestihälytyksen, jos seurattavien kohteiden arvot ylittävät asetettuja raja-arvoja.</p> <p>Lopputuloksena oli toimeksiantajan tarpeet täyttävä järjestelmä, jolla voidaan seurata laitetilan ilmaston sekä keskeytymättömän virransyötön tilannetietojen muutoksia SNMP:llä tai graafisella käyttöliittymällä. Toimeksiantajalle luotiin asennuspaketti, jolla järjestelmän voi toteuttaa uudelleen osittain tai kokonaan toiseen kohteeseen.</p>		
Avainsanat (asiasanat) Raspberry Pi, SNMP, PHP, Cacti, seuranta, valvonta, monitorointi, 1-Wire, hälytysjärjestelmä, tietokanta, UPS		
Muut tiedot		

Author(s) Holmén, Joonas	Type of publication Bachelor's thesis	Date 4/2016
	Number of pages 96	Language of publication: Finnish
		Permission for web publication: x
Title of publication Remote monitoring of server room Raspberry Pi & SNMP		
Degree programme Degree Programme in Information Technology		
Supervisor(s) Mika Rantonen, Olli Väänänen		
Assigned by JAMK University of Applied Sciences, Juha Jokinen		
<p>Description</p> <p>The thesis was assigned by JAMK University of Applied Sciences. The client wanted to be able to remotely monitor the climate of their server room and the status of an uninterruptible power supply. The aim was to create a system that allows the current status to be read with SNMP and comes with a graphical user interface, which can be used to monitor the changes.</p> <p>The platform for the work was Raspberry Pi 2 microcomputer, and it was set-up with open source software. The system consisted of two parts, remote query and monitoring. The remote query was implemented by collecting the status values and making them available with SNMP.</p> <p>Few script programs were created that enable the remote query. The programs gather the values and create an object subtree for them. The monitoring was implemented with Cacti-monitoring software that upkeeps the databases of collected values and provides an ability to generate graphs from changes in a given time.</p> <p>Cacti's functions were expanded with an alerting ability. The alerting could send an SMS alert if the monitored values breached the set thresholds.</p> <p>The end result was a system that fulfilled the requirements of the client. The system enables monitoring of the server room climate and the status of the uninterruptible power supply. An installation package that allows the system to be installed fully or partly again elsewhere was also created for the Client.</p>		
Keywords (subjects) Raspberry Pi, SNMP, PHP, Cacti, monitoring, 1-Wire, alerting, database, UPS		
Miscellaneous		

Sisältö

LYHENTEET.....	6
1. TAVOITTEET JA TEHTÄVÄT	8
2. TIETOPERUSTA	9
2.1. RASPBERRY PI	9
2.1.1. Alku.....	9
2.1.2. Rakenne.....	10
2.1.3. Ohjelmisto	11
2.1.4. GPIO-liitännät.....	12
2.2. 1-WIRE	13
2.2.1. Viestintä.....	14
2.2.2. Aosong Electronics Co. Ltd:n 1-Wire.....	17
2.3. LINUX.....	20
2.4. MYSQL.....	21
2.5. APACHE2.....	22
2.6. PHP	22
2.7. RRDTOOL.....	23
2.7.1. Tietokannan luominen	24
2.7.2. Arvojen kirjoittaminen tai lukeminen	26
2.8. CACTI.....	27
2.8.1. Mallit	29
2.8.2. Datat Kerääminen.....	30
2.8.3. Datat esittäminen	30
2.8.4. Liitännäiset	31
2.9. YKSILÖINTITUNNISTE	32
2.10. SNMP.....	33
2.10.1. Viestit.....	35
2.10.2. SNMP viesti kehys	36
2.11. KESKEYTYMÄTÖN VIRRANSYÖTTÖ	37

3. JÄRJESTELMÄN KUVAUS	38
3.1. RASPBERRY PI 2	40
3.2. ANTURIT	40
4. RPI-ILMASTO	42
4.1. TIEDOSTOT & RIIPPUVUUDET	44
4.2. SKRIPTIT	45
4.2.1. INSTALL.....	45
4.2.2. Ohjaus.....	46
4.2.3. Kerääjät	46
4.2.4. SNMP laajennus.....	47
4.2.5. Nollaaja.....	47
4.3. KONFIGURAATIOTIEDOSTO.....	48
4.4. MUUT TIEDOSTOT.....	48
5. ETÄKYSELY.....	49
5.1. ANTUREIDEN KYTKENTÄ	49
5.2. NOUDETTAVAT KOHTEET	51
5.2.1. Ilmasto.....	51
5.2.2. UPS-järjestelmän tila	52
5.2.3. Muistikortin I/O.....	54
5.3. RPI-ILMASTON ASENTAMINEN	55
5.4. SNMP-AGENTTI	57
6. VALVONTA	60
6.1. MOKKULAN KÄYTTÖÖNOTTO	60
6.2. CACTIN ASENTAMINEN	61
6.3. CACTIN ASETUKSET.....	63
6.4. CACTIN LAAJENTAMINEN	66
6.5. KOHTEIDEN LISÄÄMINEN	71
6.5.1. Laite	71
6.5.2. Seurattava asia/olio.....	72

6.5.3. Kuvio	72
6.6. NÄKYMÄN MUOKKAAMINEN	73
6.7. THOLDIN KÄYTTÄMINEN	75
6.7.1. Perusasetukset	75
6.7.2. Raja-arvon luominen	77
6.7.3. Tholdin näkymä	78
7. POHDINTA.....	79
7.1. TAVOITTEET JA TULOKSET	79
7.2. KEHITYSIDEOITA	82
LÄHTEET	83
LIITTEET	86

Kuviot

Kuvio 1 NOOBS:in asennus näkymä	12
Kuvio 2 Raspberry Pi:n GPIO-väylä.....	13
Kuvio 3 1-Wire 1- ja 0-bitti signaalit.....	15
Kuvio 4 1-Wire, nollaus ja läsnäolo viestit	15
Kuvio 5 Aosong anturin data paketin sisältö.....	18
Kuvio 6 Aosong isännän ja anturin viestinnän aloitus	19
Kuvio 7 Aosong anturin bittien siirto signaalit	19
Kuvio 8 PHP-skripti HTML:n sisällä	23
Kuvio 9 RRDToolilla luotuja kuvioita	27
Kuvio 10 Cactissa piirrettyjä kuvioita	28
Kuvio 11 Cactin toimintaperiaate.....	29
Kuvio 12 Tree View Cactissa	31
Kuvio 13 Cactin liitännäisten hallintanäkymä	32
Kuvio 14 Yksilöintitunniste puu.....	32
Kuvio 15 SNMP verkko	34

Kuvio 16 SNMP:n viestit	35
Kuvio 17 SNMP kehys	36
Kuvio 18 UPS järjestelmän kaavio	37
Kuvio 19 Järjestelmän osat	39
Kuvio 20 Antureiden pinnien liitäntä sekä Raspberry Pi:n GPIO-väylä	41
Kuvio 21 Rpi-ilmaston OID puu	42
Kuvio 22 Rpi-ilmaston kulkeminen SNMP:llä	43
Kuvio 23 INSTALL-skriptin help tuloste	45
Kuvio 24 rpi-ilmaston vihje viesti	46
Kuvio 25 Muutaman kerääjäskriptin toiminta	46
Kuvio 26 SNMP laajennus-skriptin toiminta	47
Kuvio 27 Ote rpi-ilmaston manpagesta	48
Kuvio 28 Järjestelmän kotelo ja anturit	49
Kuvio 29 Järjestelmän liittimien suunnitelma	50
Kuvio 30 Antureiden liittimet	51
Kuvio 31 DS18B20 anturin viesti	51
Kuvio 32 Apcupsd status tuloste	53
Kuvio 33 DS18 virheiden SNMP haku sanalla	57
Kuvio 34 Cactin kirjautumis ikkuna	63
Kuvio 35 Cactin rrd tietokantojen arkistojen astukset	63
Kuvio 36 Poller asetusten asettaminen cactiin	64
Kuvio 37 Cactin Data templateiden asetukset	65
Kuvio 38 Cactin Graph Templaten asetukset	66
Kuvio 39 Tekstiviesti hälytysten asetukset Tholdissa	68
Kuvio 40 Tekstiviesti hälytyksen koeastaminen	70
Kuvio 41 Laite asetukset Cactissa	71
Kuvio 42 Seurattavan kohteen lisääminen Cactiin	72
Kuvio 43 Monitorointinäkymä Cactissa	73
Kuvio 44 Cactin Graph-näkymän muokkaus	74
Kuvio 45 Cactin muokattu Graph-näkymä	74
Kuvio 46 Tholdin globaalit asetukset	75

Kuvio 47 Laitteen saatavuuden seuranta asetukset Cactissa	76
Kuvio 48 Raja-arvojen luominen Tholdiin	77
Kuvio 49 Raja-arvon asetukset	77
Kuvio 50 Tholdin käyttöliittymä	78
Kuvio 51 Toimeksiantajan rpi-ilmasto asetukset	80
Kuvio 52 Toimeksiantajalle asennetun rpi-ilmaston alipuu kuljettuna	80

Taulukot

Taulukko 1 Raspberry Pi versioiden vertailu	11
Taulukko 2 1-Wire ROM komennot	16
Taulukko 3 1-Wire orjalaitteen DS18B20 toiminta komennot.....	17
Taulukko 4 Cactin rrd tietokantojen arkistojen asetusten ajanjaksot	64

Liitteet

Liite 1 rpi-ilmasto manpage.....	86
Liite 2 Rpi-ilmaston MIB-moduuli.....	90

Lyhenteet

CF	Consolidation Function
CRC	Cyclic Redundancy Check
DS	Data Source
DST	Data Source Type
EPS	Encapsulated PostScript
GNU	GNU's Not Unix
GPIO	General Purpose I/O
HDMI	High Definition Multimedia Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ID	IDentification number
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU-Telecommunication standardization sector
LAMP	Linux, Apache, Mysql, PHP
MIB	Management Information Base
NCSA	National Center for Supercomputing Applications
NOOBS	New Out Of the Box Software
OID	Object IDentifier
PC	Personal Computer
PHP	PHP: Hypertext Preprocessor
PNG	Portable Network Graphics
RAM	Random-Access Memory
ROM	Read-Only Memory
RRA	Round Robin Archive
RRD	Round Robin Database
SMI	Structure of Management Information

SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
TCP/IP	Internet Protocol Suite
UDP	User Datagram Protocol
UPS	Uninterruptible Power Supply
USB	Universal Serial Bus
WWW	World Wide Web
XML	eXtensible Markup Language

1. Tavoitteet ja tehtävät

Työn tavoitteena oli toteuttaa Jyväskylän ammattikorkeakoulun Dynamontoimipisteessä löytyvän Spidernet-laitetilojen etäseuranta, joka mahdollistaa laitetilän lämpötilojen, ilmankosteuden sekä keskeytymättömän virransyötön tilannetietojen seurannan. Työn edetessä järjestelmään lisättiin graafinen monitorointi käyttöliittymä sekä hälytysjärjestelmä. Käyttöliittymällä seurattavia arvoja kyettiin seuraamaan verkkoselaimella sekä asettamaan raja-arvoja hälytysjärjestelmään. Hälytysjärjestelmään lisättiin toiminta, joka sallii hälytysten lähettämisen tekstiviestillä.

Seurantajärjestelmä on toteutettava pienellä budjetilla käyttäen yksinkertaisia pienlaitteita sekä avoimen lähdekoodin ohjelmistoja ja standardiprotokollia. Järjestelmän on pidettävä lokitietokantaa seurattavien arvojen muutoksista lyhyeltä ajalta sekä kysyttäessä antaa laitetilän senhetkiset arvot. Etäkysely täytyy olla toteutettuna ja saatavilla SNMP:llä. Toteutus täytyy dokumentoida ja tehdä mahdolliseksi toteuttaa uudelleen toimeksiantajan toimesta.

Järjestelmän alustaksi määriteltiin Raspberry Pi 2 -pientietokone, jota varten on huomattavat määrät laajasti saatavilla olevaa aineistoa. Seurattavat anturit kytketään Raspberry Pi:hin, sekä sille asennetaan käyttöliittymä, jolla kerättyjä tietoja voidaan katsoa ja muutoksia seurata. Koska järjestelmää ei toteuteta valmiilla ja suljetulla tuotteella, järjestelmää on tulevaisuudessa mahdollista laajentaa tai uudelleenkäyttää muissa tehtävissä.

2. Tietoperusta

2.1. Raspberry Pi

Raspberry Pi on sarja yhdenpiirilevyn pientietokoneita, jotka kehitettiin tarkoituksena saada edullinen tietokone elektroniikan ja tietotekniikan opiskelu- ja opetuskäyttöön. Pienestä koostaan huolimatta Raspberry Pi kykenee ajamaan ja toimimaan kuten mikä tahansa yleiskäyttöinen tietokone tukien muun muassa internetin selaamista, videoiden toistamista sekä toimistotyökalujen käyttöä.

Perustietokoneominaisuuksien lisäksi Raspberry Pi:llä on useita käyttäjän ohjelmoitavissa olevia liitäntäpisteitä, joihin käyttäjät voivat asentaa omia laitteita tai komponentteja. Ohjelmoitavissa olevat liittimet sekä perustietokoneen toiminnot ovat tehneet Raspberry Pi:stä erittäin suosittu harrastelijoiden parissa sekä yhden suosituimpia esineiden internet -alustoja (The Making Of Pi n.d; What is Raspberry Pi? n.d.)

2.1.1. Alku

Työskennellessään Cambridgen yliopiston tietotekniikanlaboratoriossa Eben Upton, Rob Mullins, Jack Lang ja Alan Mycroft, huomasivat nuorten tietotekniikan ja tietokone tuntemuksen olevan laskussa sekä yhä harvemman hakevan korkeakoulutukseen tietojenkäsittelytieteiden alalle. 2000-luvulle siirtyessä korkeakoulutukseen hakevilla opiskelijoilla tietotekninen kokemus rajoittui yksinkertaisen verkkosivun suunnitteluun tai tekstinkäsittelyyn, kun 1990-luvulla useat hakijat olivat harrastusten puolesta keränneet kokemusta ohjelmoinnista. (The Making Of Pi n.d.)

Tietotekniikan koulutus rajoittautui tunteihin toimistotyökalujen käytössä ja kirjoittamisessa, jättäen ohjelmoinnin sekä tietotekniikan tiedon ja taidon täysin ilman huomiota. Toteutettu koulutus ei myöskään herättänyt lapsissa tai nuorissa kiinnostusta tietotekniikkaa kohtaan. Koti-PC:t ja pelikonsolit olivat sivuuttaneet

Amigat, Spectrum ZX:t ja Comondore 64–tietokoneet, joilla aikaisemmat sukupolvet olivat oppineet ohjelmoimaan. (The Making Of Pi n.d.)

Nykyaikaiset tietokoneet olivat kehittyneet arvokkaiksi ja vaikeaselkoisiksi, joten ohjelmointi ja kaikenlainen kokeilu niin vanhempien toimesta kotona kuin opettajien toimesta koulussa usein kiellettiin. Vaikuttaakseen tähän miehet alkoivat suunnittelemaan pientietokonetta, joka olisi hinnaltaan pieni, mutta kuitenkin ominaisuuksiltaan tarpeeksi monipuolinen ja tehokas, jotta tietotekniikan eri osa-alueita voidaan tutkia ja opiskella. Vuoteen 2008 mennessä mobiililaitteissa käytetyt prosessorit olivat tarpeeksi tehokkaita ja riittävän edullisia, että projekti oli mahdollista toteuttaa. Pete Lomas sekä David Braben liittyivät projektiin mukaan ja Raspberry Pi Foundation perustettiin, tehtävänään tietotekniikan ja tietojenkäsittelytieteiden koulutuksen edistäminen, sekä Raspberry Pi - pientietokoneen kehittäminen. (The Making Of Pi n.d.)

2.1.2. Rakenne

Raspberry Pi on rakennettu Broadcomin BVM2835-Järjestelmäpiirin ympärille. Piiri sisältää suorittimen, grafiikkaprosessorin, muistin, digitaalisen signaaliprosessorin sekä USB-portin(Universal Serial Bus). Tällä hetkellä laitehoiltaan tehokkain versio on helmikuussa 2015 julkaistu Raspberry Pi 2. Uusi Raspberry Pi 2 sisälsi uudemman BVM2836-järjestelmäpiirin, joka kasvatti suorittimen tehon 700 MHz kellotaajuksista yhdenytimen prosessorista neliytimiseen 900 MHz kellotaajuuksiseen prosessoriin sekä tuplasi muistin määrän 512 megatavusta yhteen gigatavuun. (RPI Hardware n.d.)

Marraskuussa 2015 julkaistiin myös erittäin pieni Raspberry Zero niille käyttäjille, joille normaalin Raspberry Pi:n hinta oli korkea tai tarvitsivat erittäin pienen laitteen. Raspberry Zero käyttää vanhempaa BVM2835-järjestelmäpiiriä ja on tehojensa puolesta hyvin samankaltainen kuin vanhemmat Raspberry Pi:t, mutta suorittimen kellotaajuus on nostettu yhteen gigahertsiin ja kaikki ylimääräiset komponenttien osat sekä liittimet on jätetty asentamatta tai vaihdettu pienempiin. Zeron piirilevyn

koko on saatu lähes puoleen verrattuna aikaisempaan ”kokonaiseen” piirilevyyn. Taulukossa 1 on listattu muutamia kohtia eri Raspberry Pi versioiden eroista (RPI Hardware n.d; RPi Hardware History n.d).

Taulukko 1 Raspberry Pi versioiden vertailu (RPI Hardware History n.d.)

Hinta	Raspberry Pi	CPU	GPU	RAM	USB port	GPIO	Virta max	Koko
25\$	Model A	700Mhz	700Mhz	256MB	1 kpl	26 kpl	300mA	85,60mm x 56,5mm
20\$	Model A+	700Mhz	700Mhz	256MB	1 kpl	40 kpl	200mA	65mm x 56,5mm
35\$	Model B	700Mhz	700Mhz	512MB	2 kpl	26 kpl	700mA	85,60mm x 56,5mm
25\$	Model B+	700Mhz	700Mhz	512MB	4 kpl	40 kpl	600mA	85,60mm x 56,5mm
35\$	2	4x900Mhz	700Mhz	1GB	4 kpl	40 kpl	800mA	85,60mm x 56,5mm
5\$	Zero	1000Mhz	700Mhz	512MB	1 kpl	40 kpl	160mA	65mm x 30mm

2.1.3. Ohjelmisto

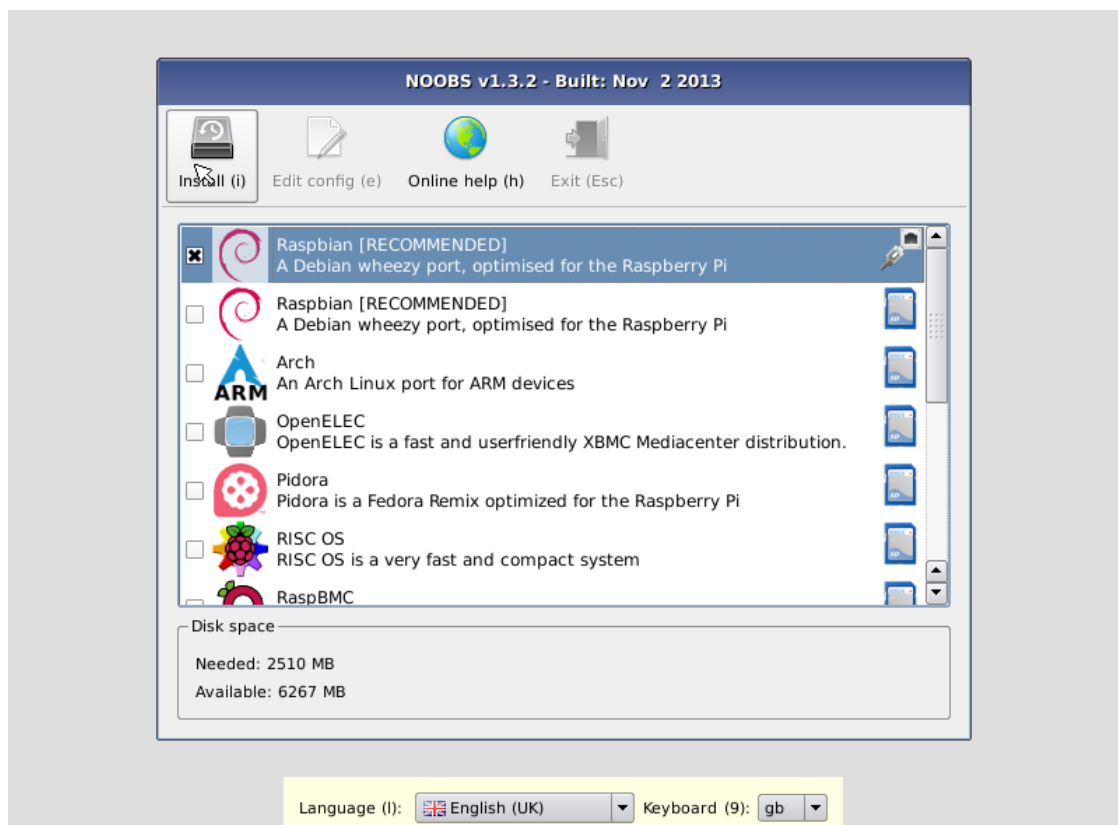
Raspberry Pi Foundation ylläpitää Linux Debianista haarautettua käyttöjärjestelmää nimeltään Rasbian. Käyttöjärjestelmä on vapaasti saatavilla yhteisön verkkosivuilta. Raspberry Pi:n opetukseen suunnatun luonteen vuoksi on laitteen käyttöönottamisen yritetty rakentaa mahdollisen helpoksi valmiiden asennuspakettien muodossa. Yhteisön sivuilla on myös linkkejä kolmansien osapuolien Linux sekä Microsoftin Windows 10 IoT -käyttöjärjestelmien sivuille (Downloads n.d; Rasbian n.d.)

Yhteisö ylläpitää Rasbian-käyttöjärjestelmän lisäksi NOOBS (New Out Of the Box Software) pakettia, joka on muistikortille siirrettävä tiedostopaketti. NOOBS sisältää automaattisia asennusohjelmia, joilla Raspberry Pi:n käyttöönottoa yritetään tehdä erittäin helpoksi myös käyttäjille, joilla ei ole kokemusta Linuxin käytöstä tai asentamisesta. Kun NOOBS on siirretty muistikortille ja asetettu Raspberry Pi:hin, käynnistyksen yhteydessä käyttäjälle tulee näkyviin kuvion 1 näkymä. Käyttäjän on mahdollista asentaa Rasbian tai antaa NOOBS:n ladata ja asentaa jokin vaihtoehtoinen käyttöjärjestelmä internetistä. NOOBS:n kautta asennettavia käyttöjärjestelmiä ovat:

- Raspbian
- Pidora

- OpenELEC
- OSMC
- RISC OS
- Arch Linux

Asennuksen yhteydessä käyttäjältä kysytään eri asetuksia, kuten käyttäjän salasanaa ja mihin käyttöliittymään Raspberry Pi käynnistetään. (NOOBS n.d.)

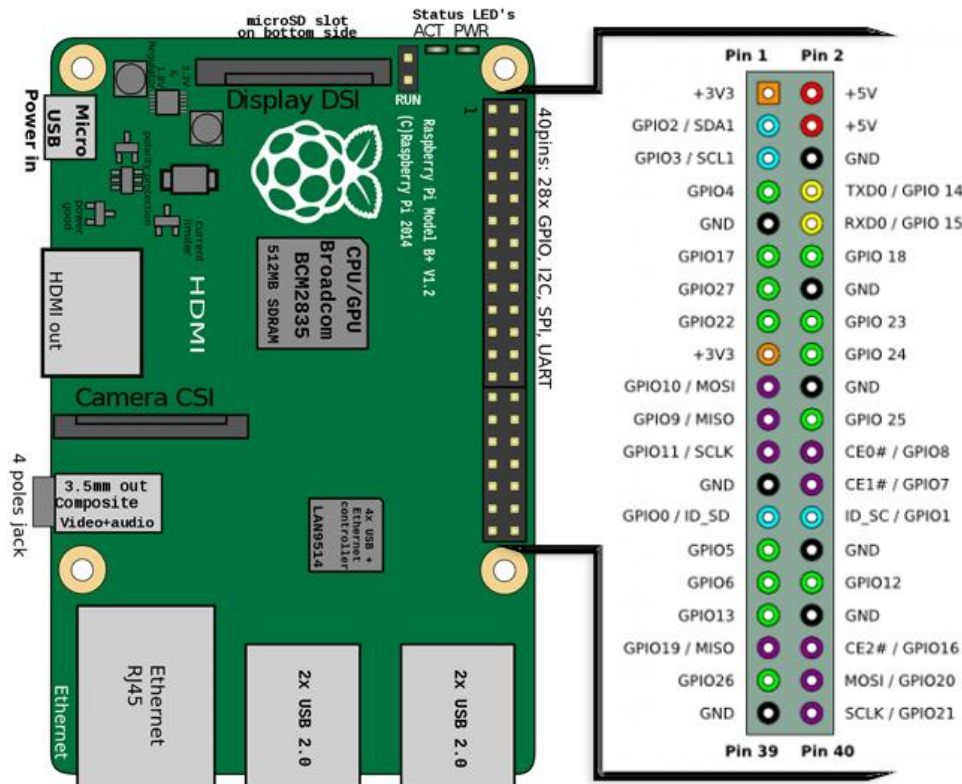


Kuvio 1 NOOBS:in asennus näkymä (NOOBS n.d.)

2.1.4. GPIO-liitännät

Yleisten laiteliittimien kuten USB, ethernet ja HDMI:n (High Definition Multimedia Interface) lisäksi, Raspberry Pi:hin on rakennettu väylä yleiskäyttöisiä I/O (GPIO, General-purpose input/output) liitin pinnejä, jotka mahdollistavat erilaisten elektroniikka komponenttien tai laitteiden liittämisen ja ohjaamisen. GPIO:t ovat käyttäjän ohjelmoitavissa vastaanottavaksi tai lähettäväksi. Alkuperäisissä Model A ja

Model B versioissa GPIO-väylä on 26 pinninen, uudemmissa versioissa GPIO-väylän koko on kasvatettu 40 pinniin. GPIO-väylässä on 28 ohjelmoitavissa olevaa GPIO-pinniä, kaksi +5 ja kaksi +3,3 Voltin jännitelähde-pinniä sekä kahdeksan maa-pinniä. Kuviossa 2 näkyy Raspberry Pi Model B+:n GPIO-väylän pinnien järjestys sekä vaihtoehtoinen tila. (GPIO n.d.)



Kuvio 2 Raspberry Pi:n GPIO-väylä (RPi Low-level peripherals n.d.)

2.2. 1-Wire

1-Wire on Dallas Semiconductor Corp:in (nykyään Maxim Integrated) kehittämä data-väylä, joka toimii ja rakentuu nimensä mukaisesti yhdellä signaalilla. Väylällä tapahtuva kommunikointi on asynkronoitua ja vuoro-suuntaista. Väylälle voidaan asettaa useita laitteita sarjaan, jotka muodostavat 1-Wire verkon. Laitteet yhdistetään toisiinsa käyttäen kahta johdinta, data ja maa. Toimintavirta otetaan myös data-johtimesta. Laitteisiin on asennettu kondensaattori, jolla laite pitää

itsensä toiminnassa väylän ollessa alatilassa. Mikään ei tosin estä antamasta laitteille toimintavirtaa 1-Wire verkon ulkopuolelta. Verkko rakentuu verkkoa ohjaavasta isäntälaitteesta (master) sekä väylälle asetetuista orjalaitteista (slave). Orjalaitteiden ROM-muistiin (Read-Only Memory) on kirjattu uniikit ID:t (IDentification number), joilla isäntälaitte kykenee tunnistamaan yksittäiset laitteet verkossa. 1-Wiren siirtonopeus on vakiona 16,3 kbit/s, mutta toimintasäde on pitkä (300m). Sitä käytetäänkin enimmäkseen useiden, mutta pienien laitteiden ohjaamiseen (esim. anturit tai tunnistimet). (Overview of 1-Wire Technology and its use 2008.)

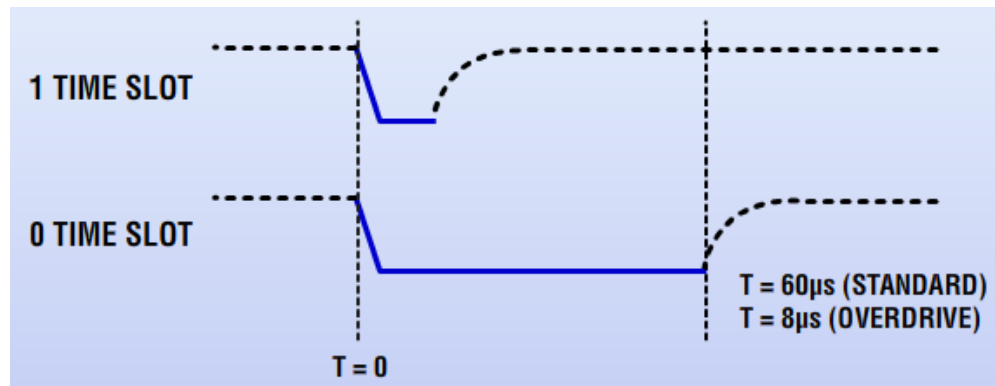
2.2.1. Viestintä

1-Wiren signaaliväylä on joutilaana ylätilassa, verkossa olevien laitteiden on kyettävä vetämään signaali alas voidakseen kommunikoida 1-Wire väylällä. Kaikki liikenne väylällä tapahtuu isännän aloitteesta ja bitti kerrallaan. Bittien signalointi väylällä tapahtuu jakamalla sen tila aikaväleihin. Aikavälin pituus on normaalitilassa 60 mikrosekuntia, mutta jotkin laitteet tukevat nopeampaa 8 mikrosekunnin Overdrive-viestintää. (AVR318: Dallas 1-Wire master 2004.)

Bittien siirtämistä varten on isännän puolesta kolme eri signaalia: *kirjoita 1*, *kirjoita 0* tai *lue*. Signaalit, joilla isäntä lähettää bitin *1* tai *0*, on esitetty kuviossa 3. Signaaleja toistetaan aikaväli toisensa jälkeen, kunnes viesti on siirretty kokonaan.

- Kirjoittaakseen bitin *1* isäntä vetää väylän alas 1-15 mikrosekunniksi ja päästää sen takaisin joutilaaseen ylätilaan aikavälin loppuun.
- Kirjoittaakseen bitin *0* isäntä vetää väylän alas koko aikavälin ajaksi.
- Lukiessaan orjalaitetta isäntä vetää signaalin alas 1-15 mikrosekunniksi ja päästää sen takaisin joutilaaseen ylätilaan. Jos orjalaite estää väylän nousemisen pitämällä sen vedettynä alas, orjalaite lähettää bitin *0*, jos orjalaite antaa väylän nousta joutilaaseen ylätilaan, lähettää se bitin *1*.

Signaalit *kirjoita 1* ja *lue* ovat isännän puolesta samat. Orjalaitteen sen hetkinen tila määrittää onko signaali lukemista vai bitin 1 kirjoitus. (AVR318: Dallas 1-Wire master 2004.)

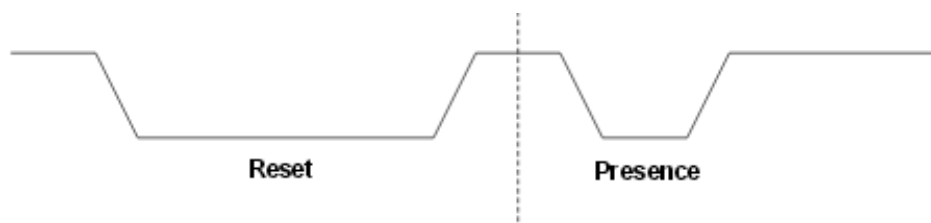


Kuvio 3 1-Wire 1- ja 0-bitti signaalit (1-Wire tutorial n.d.)

1-Wire verkossa viestintä on kolmiosainen tapahtuma:

1. Väylätila nollataan ja orjalaitteet lähettävät läsnäolo pulssin, kts. kuvio 4.
2. Orjalaitteen valinta. Isäntä lähettää väylälle ROM-komennon.
3. Toiminta komento. Isäntä lähettää orjalaite kohtaisen komennon.

Ensimmäisessä vaiheessa väylän nollauksella verkon orjalaitteet saadaan synkronoitua odottavaan tilaan. Väylätila nollataan isännän toimesta vetämällä väylä loogiseen nollaan kahdeksan aikavälin ajaksi, eli 480 mikrosekunniksi. Yhden aikavälin sisällä orjalaitteet ilmoittavat läsnäolonsa vetämällä väylän nollaan ~1-2 aikavälin ajaksi. Koska kaikki orjalaitteet vetävät väylän alas yhtä aikaa, isäntä tietää ainoastaan verkossa olevan ainakin yksi orjalaite. (1-Wire Tutorial n.d.)



Kuvio 4 1-Wire, nollaus ja läsnäolo viestit (AVR318: Dallas 1-Wire master, 2004)

Toisessa vaiheessa isäntä lähettää väylälle ROM-komennon. Jokaisella 1-Wire orjalaitteella on globaali ROM-muistiin kirjattu 16-bittinen tunniste. Tunniste rakentuu kolmesta osasta. Ensimmäinen 8 bittiä tunnistaa laiteperheen, seuraavat 48 bittiä osoittaa laitteen sarjanumeron ja viimeiset 8 bittiä on ensimmäisestä 58 bitistä laskettu tarkistussumma (CRC, Cyclic Redundancy Check). ROM-komennoilla tarkoitetaan pientä joukkoa komentoja, jotka tavalla tai toisella toimivat tämän tunnisteiden kanssa. Olemassa olevat ROM-komennot löytyvät taulukosta 2. (AVR318: Dallas 1-Wire master 2004.)

Taulukko 2 1-Wire ROM komennot (AVR318: Dallas 1-Wire master, 2004)

Command	Code	Usage
READ ROM	33H	Identification
SKIP ROM	CCH	Skip addressing
MATCH ROM	55H	Address specific device
SEARCH ROM	F0H	Obtain IDs of all devices on the bus
OVERDRIVE SKIP ROM	3CH	Overdrive version of SKIP ROM
OVERDRIVE MATCH ROM	69H	Overdriver version of MATCH ROM

- **Read ROM:** Jos väylällä on yksi orjalaite, käskee orjalaitteen ilmoittaa 64-bittisen tunnisteensa.
- **Skip ROM:** Orjalaitteen valinta jätetään väliin, siirrytään välittömästi toiminta komennon antamiseen. Voidaan käyttää ohjaamaan yhtä orjalaitetta, jos väylällä ei ole kuin yksi laite. Jos väylällä on useita orjalaitteita, voidaan käyttää antamaan komento kaikille laitteille yhtäikaa.
- **Overdriver Skip ROM:** Sama kuin edellinen, lisäksi asetetaan väylä nopeaan Overdrive tilaan.
- **Match ROM:** Valitaan orjalaite, jonka kanssa halutaan kommunikoida. Muut orjalaitteet siirtyvät toimettomaan tilaan.
- **Overdrive Match ROM:** Valitaan orjalaite ja asetetaan orjalaite nopeaan viestintä tilaan.
- **Search ROM:** Aloitetaan verkossa olevien orjalaitteiden tunnistaminen.

Viimeisessä vaiheessa kommunikointi tapahtumaa isäntä lähettää orjalaitteelle toimintakomennon. Toimintakomennot ovat orjalaite kohtaisia ja riippuvat sen tyyppistä ja toiminnasta. Usein laitteita komennetaan suorittamaan jokin tehtävä, kirjoittamaan jotain muistiinsa tai lukemaan ja ilmoittamaan jokin tieto muististaan. Esimerkkinä taulukossa 3 on listattu lämpötila anturi DS18B20:lle annettavia komentoja. (1-Wire Tutorial n.d.)

Taulukko 3 1-Wire orjalaitteen DS18B20 toiminta komennot (DS18B20 Programmable Resolution 1-Wire Digital Thermometer 2008)

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED
TEMPERATURE CONVERSION COMMANDS			
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).
MEMORY COMMANDS			
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H , T_L , and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.
Copy Scratchpad	Copies T_H , T_L , and configuration register data from the scratchpad to EEPROM.	48h	None
Recall E^2	Recalls T_H , T_L , and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.

2.2.2. Aosong Electronics Co. Ltd:n 1-Wire

Aosong Electronicsin valmistamat anturit käyttävät 1-Wiren tapaista kommunikointitapaa. Anturit toimivat samaan tapaan kahden johtimen kanssa, sekä sisältävät sisäisen kondensaattorin ylläpitääkseen anturin toimintaa silloin kun väylä on alhaalla. Toisin kuin 1-Wire, joka on kokonainen kahdensuuntainen protokolla, Aosongilla isännällä ja orjalaitteella on vain kaksi viestiä: Isännän *start*-signaali, ja

anturin *datan lähetys*. Väylä ei tue kuin yhtä orjalaitetta, sillä Aosongin anturit tai data protokolla ei sisällä mitään tunnistusmetodia. Väylällä oletetaan aina olevan ainoastaan yksi laite. Laitteet eivät kommunikoi keskenään väylän käytöstä, vaan olettavat väylän olevan käytettävissä aina lähettäessään oman viestinsä. (Thomas Liu n.d.)

DHT-22 mallinen anturi lähettää kosteus (0 - 100%) ja lämpötila tiedon (-100 - 100°C) 40-bittisenä pakettina, jossa lähetetty arvo on kymmenkertainen oikeaan arvoon verrattuna. Data jaetaan viiteen 8-bittiseen ryhmää. Ensimmäiset kaksi 8-bitin ryhmää ovat kosteuden arvo 16-bittinä, alkaen merkitsevimmästä bitistä. Seuraavat kaksi 8-bitin ryhmää ovat lämpötila arvo 16-bittinä, jos lämpötilan ensimmäinen lähetetty bitti on 1, lämpötila-arvo on negatiivinen. Viimeinen 8-bittiä on tarkistussumma, joka on aikaisemmat 8-bitin ryhmät summattuna yhteen. Kuviossa 5 on esimerkki yhdestä mahdollisesta viestistä, jonka anturi voisi lähettää ja kuinka sen arvot ja tarkistussumma lasketaan. (Thomas Liu n.d.)

◎ Single-bus data calculation example

Example 1: 40 Data received:

<u>0000 0010</u>	<u>1001 0010</u>	<u>0000 0001</u>	<u>0000 1101</u>	<u>1010 0010</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Calculate:

$0000\ 0010 + 1001\ 0010 + 0000\ 0001 + 0000\ 1101 = 1010\ 0010$ (Parity bit)

Received data is correct:

humidity: $0000\ 0010\ 1001\ 0010 = 0292\text{H}$ (Hexadecimal) $= 2 \times 256 + 9 \times 16 + 2 = 658$
 \Rightarrow Humidity = 65.8%RH

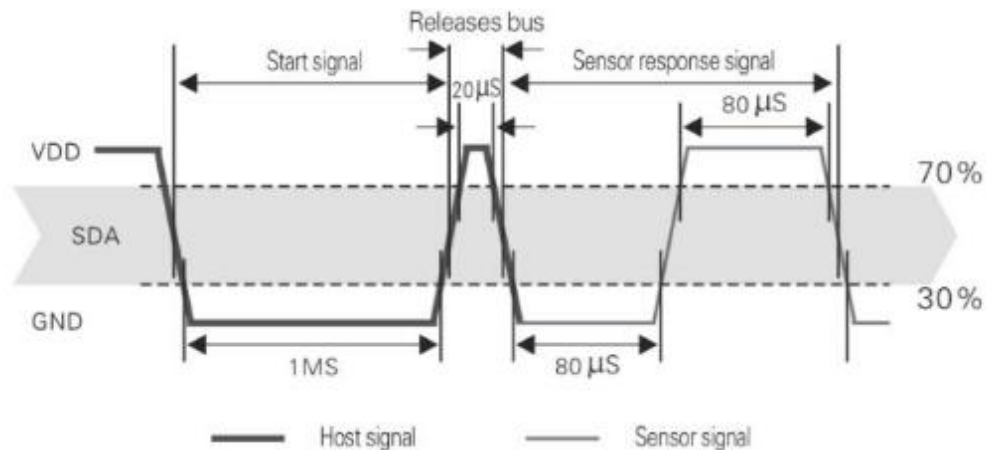
Temp.: $0000\ 0001\ 0000\ 1101 = 10\text{DH}$ (Hexadecimal) $= 1 \times 256 + 0 \times 16 + 13 = 269$
 \Rightarrow Temp. = 26.9°C

Kuvio 5 Aosong anturin data paketin sisältö (AM2302 Product Manual n.d)

Aosongin 1-Wire väylän toiminta:

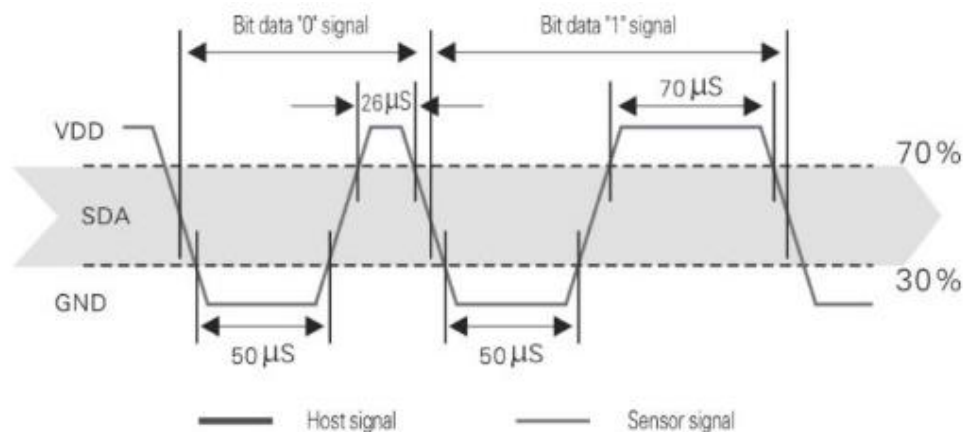
- Isäntä kääntää anturia lähettämään sen hetkisen datan vetämällä väylän alas vähintään yhdeksi millisekunniksi.

- Anturi ilmoittaa olemassa olonsa vetämällä väylän alas 80 mikrosekuniksi, jonka jälkeen se jättää väylän ylös 80 mikrosekuniksi. Kuviossa 6 näkyy isännän väylän alas veto sekä anturin ilmoitus ennen datan lähettämistä.



Kuvio 6 Aosong isännän ja anturin viestinnän aloitus (AM2302 Product Manual n.d.)

- Tämän jälkeen anturi lähettää datansa yksi bitti kerrallaan. Ennen jokaista bittiä anturi vedetään alas 50 mikrosekuniksi, jonka jälkeen se päästetään ylös. Ylätila pidetään 26–28 mikrosekuntia jos lähetettävä bitti on 0 tai 70 mikrosekuntia jos lähetettävä bitti on 1. Molempien bittien lähetys on näkyvissä kuviossa 7.



Kuvio 7 Aosong anturin bittien siirto signaalit (AM2302 Product Manual n.d.)

- Kun data on lähetetty, anturi vetää väylän viimeisen kerran alas 50 mikrosekuniksi. (AM2302 Product Manual n.d.)

2.3. Linux

Linux on Unixin kaltaisten Linux-ytimeistä (kernel) sekä GNU-projektin ohjelmista rakentuvien käyttöjärjestelmien perhe, tunnetaan myös nimellä GNU/Linux. Linux-käyttöjärjestelmä rakentuu enimmäkseen avoimesta lähdekoodista ja sen kehitys tapahtuu yhteistyössä useiden eri tahojen välillä. Linuxista on saatavilla useita eri jakeluversioita, joita ylläpitää erilaiset yhteisöt tai yritykset ja on käytettävissä lähes kaiken tyyppisillä tietokonelaitteilla. (What is Linux n.d.)

Vaikka Linuxin katsotaan edustavan yli 25 miljardin US dollarin arvoisia markkinoita, se ei ole saanut huomattavaa osaa kotitietokoneiden käyttöjärjestelmänä (1,5 %). Se on kuitenkin vallannut laajimmassa käytössä olevan palvelin (98,3 %) ja mobiililaitte (66,2 %) käyttöjärjestelmän aseman. (StatCounter Global Stats 2016; OS Market Share 2016; Who Writes Linux 2015.)

GNU-projekti (GNU's Not Unix) sai alkunsa vuonna 1983, päämääränä kehittää 100 % vapaaseen ja avoimeen lähdekoodiin perustuva käyttöjärjestelmä. Projektin laitto alulle Richard Stallman ärsyyntyessään siitä, kuinka ohjelmoinnin ala siirtyi tietotekniikan kaupallistumisen seurauksena yhä suljetumpaan suuntaan. 1990-luvulle siirtyessä GNU sisälsi lähes kaikki käyttöjärjestelmän vaatimat ohjelmistot, ydintä lukuun ottamatta. (The GNU Project 2015.)

Linux-ytimen kehittämisen aloitti vuonna 1991 Linus Torvalds. Hän halusi itselleen käyttöjärjestelmän, jota käyttää uudella i386-tietokoneellaan. Tammikuussa 1992 Linux oli kehittynyt versioon 0.12 ja se siirrettiin GNU GPL -lisenssin alle, jolloin GNU otti ytimen omaan käyttöjärjestelmäänsä. Nykyinen Linux-ytimen versio on 4.4. Ytimen vakaita päivityksiä julkaistaan 2-3 kuukauden välein, lisäten uusia toimintoja ja tukia uusille laitteille. Ytimen kehitystyö on kiihtynyt nopeasti 2000-luvun aikana, vuodesta 2005 lähtien yli 11 800 henkilöä lähes 1200 yrityksestä ovat osallistuneet Linux-ytimen kehitykseen. (What is Linux n.d; Who Writes Linux 2015.)

2.4. MySQL

MySQL on kevyt, helppokäyttöinen ja nopeasti käyttöönotettava relaatiotietokantaohjelmisto. MySQL on erittäin suosittu tietokantaohjelmisto etenkin web-palveluiden kanssa ja sitä käyttää muun muassa Facebook, Google ja Adobe. Ohjelmiston kehitti alun perin MySQL AB, mutta ohjelmiston omistaa nykyään Oracle Corporation. MySQL saa nimensä yhden alkuperäisen kehittäjän Michael Wideniuksen tyttären, My Wildeniuksen, sekä IBM:n kehittämän SQL (Structured Query Language) -kyselykielen mukaan. (MySQL 5.7 Reference Manual 2016, 1.3.1–1.3.3.)

Relaatiotietokannassa tiedot tallennetaan erillisiin taulukoihin sen sijaan, että kaikki tieto tallennettaisiin yhteen kohteeseen. Tietokanta rakentuu taulukoiden, rivien ja sarakkeiden yhdistelmistä sekä säännöistä ja yhteyksistä, joita näiden välille rakennetaan. Tietokanta käyttää sääntöjä ja yhteyksiä varmistamaan, että tietokantaan ei pääse syntymään orpoa, vanhentunutta, monistunutta tai ristiriitaista tietoa. Käyttäjä pystyy noutamaan eri taulukoista eri yhteyksiä hyväksi käyttäen eri tietoja ja rakentamaan näistä erilaisia taulukoita. (MySQL 5.7 Reference Manual 2016, 1.3.1.)

SQL on kyselykieli, jota käytetään kommunikoidessa tietokantapalvelun ja tietokannan kanssa. SQL kielellä luodaan komento lauseita, joilla tietokantaa voidaan ohjata. SQL-kielellä voidaan:

- Noutaa tietoja.
- Määritellä ja muokata tietokantaa.
- Muokata käyttöoikeuksia.
- Virittää tietokannan talletusrakenteita.
- Ylläpitää dataa lisäämällä, muuttamalla tai poistamalla. (SQL Language Reference 2006.)

2.5. Apache2

Apache on avoimen lähdekoodin projekti, jonka tarkoituksena on kehittää ja ylläpitää web-palvelin ohjelmistoa moderneille käyttöjärjestelmille. Apache on rakennettu modulaariseksi, sen toimintaa voidaan laajentaa ja muokata palvelun ylläpitäjän tarpeiden mukaan. Laajentamisella voidaan saada Apachelle tuki eri ohjelmointikielten tulkeille tai kyky ajaa ulkopuolisia ohjelmia. Apache sai alkunsa vuonna 1998 vanhemman NCSA:n (National Center for Supercomputing Applications) luoman NCSA HTTPd -palvelinohjelmiston päivittely projektina, kun tämän kehittäminen hidastui. Nyt Apachea ylläpitää Apache Software Foundation ja on kehittynyt versioon 2.4. (What is Apache n.d.)

2.6. PHP

PHP (PHP: Hypertext Preprocessor) on avoimen lähdekoodin yleiskäyttöinen skriptauskieli, jota käytetään enimmäkseen dynaamisten web-sivujen luomisessa. PHP-skripti ajetaan WWW (World Wide Web) -palveluita tarjoavalla palvelimella. PHP-skriptiä ei näytetä web-sivun avaavalle käyttäjälle, vaan se ajetaan palvelimella ja tästä generoidaan näytettävä sivu käyttäjälle. PHP pystyy keräämään tietoja, generoimaan dynaamisia sivuja sekä lukemaan ja kirjoittamaan käyttäjän evästeitä. (PHP Manual n.d, What is PHP? n.d.)

PHP voidaan kirjoittaa omaan tiedostoon tai upottaa HTML (Hypertext Markup Language) -tiedoston sisälle. PHP-skripti erotellaan muusta tekstistä ja komennoista kirjoittamalla merkit "<?php" ja "?>" joiden väliin PHP-skripti kirjoitetaan. Esimerkiksi kuviossa 8 on PHP-skripti upotettu HTML-tiedoston sisään, joka tulostaa käyttäjälle viestin "Hi, I'm a PHP script!". (PHP Manual n.d, What can PHP do? n.d.)

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>

```

Kuvio 8 PHP-skripti HTML:n sisällä (PHP Manual n.d, What is PHP? n.d.)

PHP:ta käytetään laajasti luomaan web-sivustoja, joilla on dynaamisia toimintoja. Usein PHP:ta käyttävät sivustot lukevat/kirjoittavat tietokannoista tai ajavat muita komentoja palvelimella. Käyttäjältä voidaan kerätä tietoja lomakkeilla. Käyttäjän näkymää ja kokemusta sivustolla voidaan ohjata lukemalla ja kirjoittamalla evästeitä. PHP:lla on laaja tuki useille eri käyttöjärjestelmille, mukaan lukien Unix, Linux, Microsoft Windows ja Mac OS X. (PHP Manual n.d, What can PHP do? n.d.)

2.7. RRDTool

RRDtool (Round-Robin Database tool) on aikasarjaisen numeraalisen tiedon (esim. verkkoliikenne, lämpötilat, kuormitus) tallentamista ja tallennetun tiedon käsittelyä varten luotu ohjelmisto. Ohjelmalla luodaan tietokanta ja kerrotaan kuinka montaa tietolähdettä seurataan, sekä kuinka lähteistä kerätty tieto tallennetaan. Tallentaminen tapahtuu rengaspuiskurin tapaisesti tietokannan sisällä oleviin arkistoihin. Arkistoihin kirjoitetaan käyttäjän asettama määrä rivejä. Viimeisen rivin täyttyessä aloitetaan kirjaaminen uudelleen ensimmäisestä rivistä. Käyttäjän on myös mahdollista käyttää matemaattisia algoritmeja, joilla lukuarvoja voidaan muokata ennen kuin ne kirjataan arkistoon. (Oetiker 2015.)

Tietokannan ylläpitämisen lisäksi ohjelmistolla on mahdollista piirtää kerätyistä lukuarvoista kuvioita. RRDTool tarjoaa suuren määrän eri asetuksia, joilla käyttäjä voi vaikuttaa lähes kaikkeen mitä kuvioon generoidaan. (Oetiker 2015.)

2.7.1. Tietokannan luominen

Kun tietokanta luodaan, asetetaan sille aloitusaika (oletuksena luomishetken aika), sekä aikaväli, jonka välein arvoja kirjataan tietokantaan. Aikaväliä kutsutaan askeleeksi, step.

```
rrdtool create filename --start --step
```

Seurattavat kohteet eritellään tietokantaan tietolähteinä, Data Source (DS).

Asettaessa tietolähteitä annetaan lähteelle nimi (ds-name), lähteentyyppi (DST, Data Source Type), katkosaika (heartbeat) sekä pienimmän ja suurimman hyväksyttävän lukuarvon rajat. (Oetiker 2015.)

```
DS:ds-name:DST:heartbeat:min:max
```

- **ds-name:** Käytetään erittelemään tämän lähteen lukuarvot tietokannasta.
- **DST:** Kertoo tietolähteen tyyppin, vaikuttaa muihin asetuksiin ja tiedon käsittelyyn. Mahdollisia *DST* asetuksia ovat:
 - **GAUGE:** Yksinkertainen lukuarvojen tallentaminen.
 - **COUNTER:** Jatkuvasti kasvaville lähteille, kuten laskurit. Olettaa että lukuarvo ei koskaan pienene, paitsi kun seurattava kohde ylivuotaa. Ylivuodon tapahtuessa, kirjattaviin lukuarvoihin lisätään takaisin ylivuotanut luku.
 - **DCOUNTER:** Toimii samaan tapaan kuin COUNTER, mutta tarkoitettu liukuluvuille ja kykenee seuraamaan myös laskevia lukuarvoja.
 - **DERIVE:** Kirjaa edellisestä nykyiseen lukuarvoon kulkevan käyrän derivaatan lukuarvon.
 - **ABSOLUTE:** Lukemisen jälkeen nollaantuville laskureille.

- **COMPUTE:** "Virtuaalinen" datalähde, jota käytetään kirjaamaan muiden tietolähteiden lukuarvoihin sovellettuja laskukaavioiden tuloksia.
- **Heartbeat:** Kertoo enimmäisajan sekunteina, joka odotetaan ennen kuin arvon oletetaan olevan tuntematon.
- **Min** ja **max:** Käytetään rajaamaan hyväksyttävien lukuarvojen alue. Alueen ulkopuolelle jäävät lukuarvot kirjataan tuntemattomiksi. Asetukseksi voidaan asettaa "U", jos sitä ei halua käyttää. (Oetiker 2015.)

Viimeisenä tietokannalle määritetään arkistot (RRA, Round Robin Archive), joihin lukuarvot viime kädessä kirjataan. Jokaiseen arkistoon kirjataan kaikkien tietolähteiden antamat lukuarvot. RRA:lle asetetaan tapa, jolla lukuarvot kirjataan arkistoon (CF, Consolidation Function), virhesuhde (xff), askelmäärä (steps) sekä rivimäärä (rows).

RRA:CF xff:steps:rows

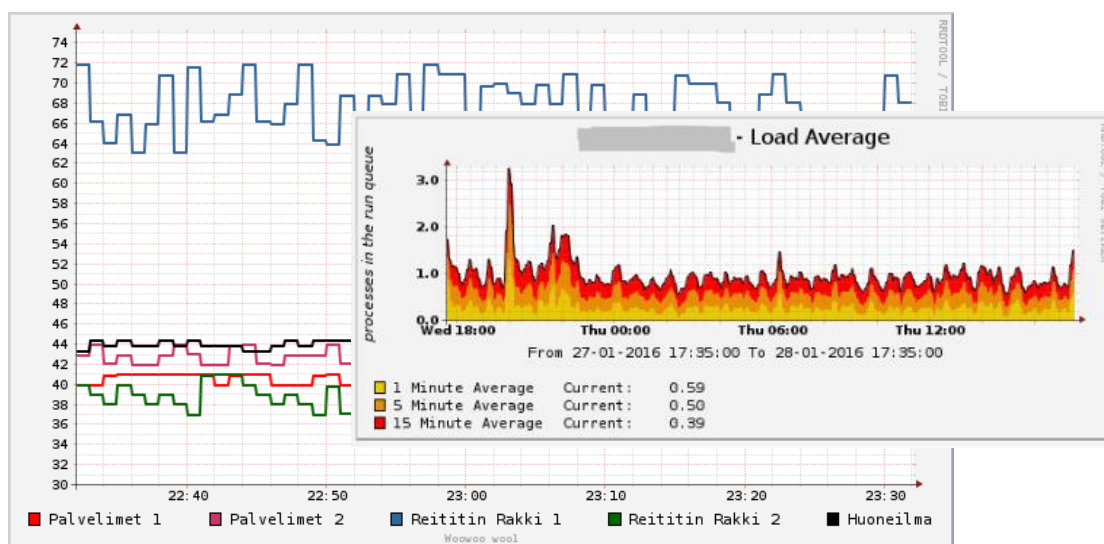
- **CF:** Kertoo kuinka kerätyt lukuarvot asetetaan arkistoon, jos arkistolle asetettu step on 1, asetuksella ei ole merkitystä. Jos arkiston step-asetus on suurempi kuin 1, asetusten toiminta on seuraava:
 - **AVERAGE.** Kerättyjen lukuarvojen keskiarvo kirjataan arkistoon.
 - **MIN.** Kerätyistä lukuarvoista pienin kirjataan arkistoon.
 - **MAX.** Kerätyistä lukuarvoista suurin kirjataan arkistoon.
 - **LAST.** Kerätyistä lukuarvoista uusin kirjataan arkistoon.
- **Xff (0-1):** Osoittaa kuinka iso osa kerätyistä arvoista saa olla tuntemattomia suhteessa tunnettuihin arvoihin, ennen kuin arvo kirjataan tuntemattomaksi.
- **Steps:** Asettaa kuinka monta askelta odotetaan, eli arvoa kerätään, ennen kuin kerätyt lukuarvot käsitellään asetetun CF:n mukaisesti ja kirjataan arkistoon.
- **Rows:** Rivimäärä, kertoo kuinka monta riviä arvoja pidetään arkistossa. (Oetiker 2015.)

2.7.2. Arvojen kirjoittaminen tai lukeminen

Tietokannan arkistoissa olevia lukuarvoja muokataan *update*-komennolla. Koska arkistot ovat luomishetkellä täytetty kokonaan, mutta arvot ovat asetettu ”*tuntemattomiksi*”, voidaan uusienkin arvojen lisääminen katsoa olevan arkiston rivillä olevan arvon muokkaamista. *Update*-komennolle annetaan aika, joka halutaan asettaa ja tämän jälkeen lukuarvot, jotka sille kohdalle arkistoihin halutaan asettaa. Asettamalla ajaksi ”*N*”, RRDTool noutaa järjestelmän sen hetkisen ajan. Lukuarvoja voidaan asettaa joko yksi tietolähde kerrallaan tai kaikkiin tietolähteisiin yhtä aikaa. (Oetiker 2015.)

Arvojen lukeminen tietokannasta on mahdollista muutamalla eri tavalla. Komennolla *xport* tietokannan voi kopioida XML (eXtensible Markup Language) -tiedostoon. Arvot voidaan lukea myös komennolla *lastupdate*, joka tulostaa tuoreimman ajankohdan jolloin arvoja on lisätty, sekä kaikkien data-lähteiden sen hetkiset arvot jonossa. Tietokannasta on myös mahdollista hakea arvoja käyttäjän valitsemalta ajankohdalta, kuten myös käyttäjän valitsemista datalähteistä. Tietokannoista voidaan myös noutaa useita arvoja käyttäjän valitseman ajankohtien välistä ja käyttäjän valitsemalla tarkkuudella. Viimeinen tapa on luoda kuvio. RRDtool kykenee luomaan muun muassa PNG- (Portable Network Graphics), SVG- (Scalable Vector Graphics) tai EPS (Encapsulated PostScript) -formaatin kuvioita. Kuvion luomiseen voidaan vaikuttaa monin tavoin. Kuviossa 9 on nähtävissä kaksi erilaista RRDToolilla luotua kuviota. Mahdollisia muokattavia asetuksia ovat muun muassa:

- Kuvion koko.
- Kuviossa käytetyt värit.
- Aikavälin valinta.
- Leimojen ja merkintöjen lisääminen.
- Arvojen muokkaaminen ennen piirtämistä.
- Tekstien ja numeroiden fontit sekä kirjainkoko. (Oetiker 2015.)

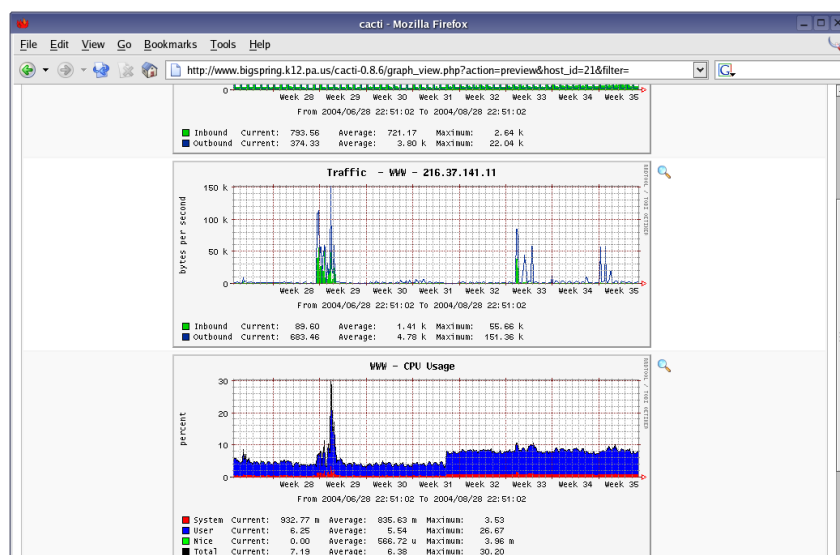


Kuvio 9 RRDToolilla luotuja kuvioita

2.8. Cacti

Cacti on PHP-kielellä kirjoitettu verkonmonitorointiohjelmisto, joka on luotu käyttöliittymäksi RRDTool -ohjelmalle. Valvonta tapahtuu tallentamalla seurattavien kohteiden arvoja round robin -tietokantaan ja piirtämällä näistä arvoista kuvioita (kuvio 10), joissa on näkyvissä viimeisin arvo sekä käyriä arvojen muutoksista valitulla ajanjaksolla. (What is Cacti n.d.)

Arvot noudetaan pääsääntöisesti käyttäen SNMP:tä, mutta Cacti kykenee ajamaan omalla isäntälaitteellaan myös skriptejä ja tallentamaan näiden tulosteita. Cacti käyttää omien asetusten ylläpitoon MySQL-tietokantaa. Asetukset sisältävät kuinka ja mistä arvoja kerätään, sekä kuinka kerättyjä arvoja käsitellään. Asetusten muokkaaminen ja kohteiden seuranta tapahtuu WWW-selaimella avattavalla käyttöliittymällä. (What is Cacti n.d.)

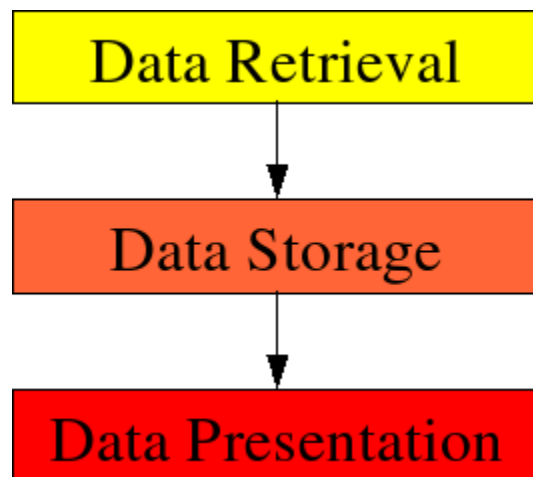


Kuvio 10 Cactissa piirrettyjä kuvia (Screen Shots n.d.)

Ian Berry aloitti Cactin kehittämisen kesäkuussa vuonna 2001, työskennellessään internet palveluntarjoajalla Yhdysvalloissa. Hän huomasi RRDToolin olevan ominaisuuksiltaan tarpeeksi laaja piirtämään ja tuottamaan monimutkaisia kuvia sekä raportteja verkon tilasta ja tapahtumista, mutta sillä ei ollut toimivaa tai käyttäjäystävällistä käyttöliittymää. Viimeisin julkaistu versio Cactista on 0.8.8f ja sen kehitystiimi on kasvanut kuuteen aktiiviseen jäseneseen. (Kundu & Lavlu 2009.)

Cactin toimintaperiaate on jaettu kolmeen pääosalueeseen, joiden yhteys toisiinsa nähtävissä kuviossa 11.

1. *Datan noutaminen.*
2. *Datan säilöminen* käyttäjän valitsemalla tarkkuudella ja tavalla round robin - tietokantaan.
3. *Datan esittäminen* käyttäjän valitsemalla tavalla piirtäen RRDToolilla. (Berry, Roman, Adams, Pasnak, Conner, Scheck & Braun 2013.)



Kuvio 11 Cactin toimintaperiaate

Datan noutamista sekä säilömistä suorittaa pollaus-ohjelma. Pollaaja ajetaan käyttäjän valitseman päivitysajan välein käyttäen crontabia. Cactille on kaksi eri pollaus-ohjelmaa. Oletus pollaaja on *cmd.php*, joka on PHP-kielellä kirjoitettu pollaaja. Toinen vaihtoehto on *spine*, joka on C-kielellä kirjoitettu pollaaja. Molemmat ohjelmat tekevät saman tehtävän, mutta *spine* on suunniteltu olemaan erittäin nopea. Jos Cactin seurattavien kohteiden määrä on erittäin suuri, eikä *cmd.php* onnistu noutamaan kaikista kohteista arvoja asetetun päivitysvälin aikana, on suotavaa vaihtaa *spineen*. (Berry ym. 2013.)

2.8.1. Mallit

Cacti käyttää malleja (*template*) helpottamaan uusien kohteiden lisäämistä ja kuvioiden luomista. Malli-tiedostot sisältävät eri oletusasetuksia, joita hyödyntäen käyttäjä voi nopeasti lisätä useita samankaltaisia kohteita tietokantaan vain muutamia yksityiskohtia muuttamalla.

Cactilla on kolmenlaisia eri malleja:

- *Host Templates*: Käytetään osoittamaan seurattavia laitteita. Malleissa on esimerkiksi käyttöjärjestelmä kohtaisia oletusasetuksia.
- *Data Templates*: Käytetään kun Cactille kerrotaan tarkasti seurattavia olioita tai kohteita.
- *Graph Templates*: Sisältää kuvioiden luomiseen ja piirtämiseen liittyviä oletusasetuksia.

Käyttäjän on mahdollista muokata olemassa olevia malleja tai luoda omia malleja. (Berry ym. 2013.)

2.8.2. Datan Kerääminen

Tiedon noutaminen eri kohteista Cactilla tapahtuu pääasiassa SNMP:llä. Cactin isäntälaitteelta voidaan arvoja kerätä myös ajamalla skriptejä, mutta ulkopuolisilta kohteilta arvojen noutaminen on mahdollista ainoastaan SNMP:llä. Cacti käyttää arvojen säilömiseen RRDToolia ja sen tietokantoja. (Berry ym. 2013.)

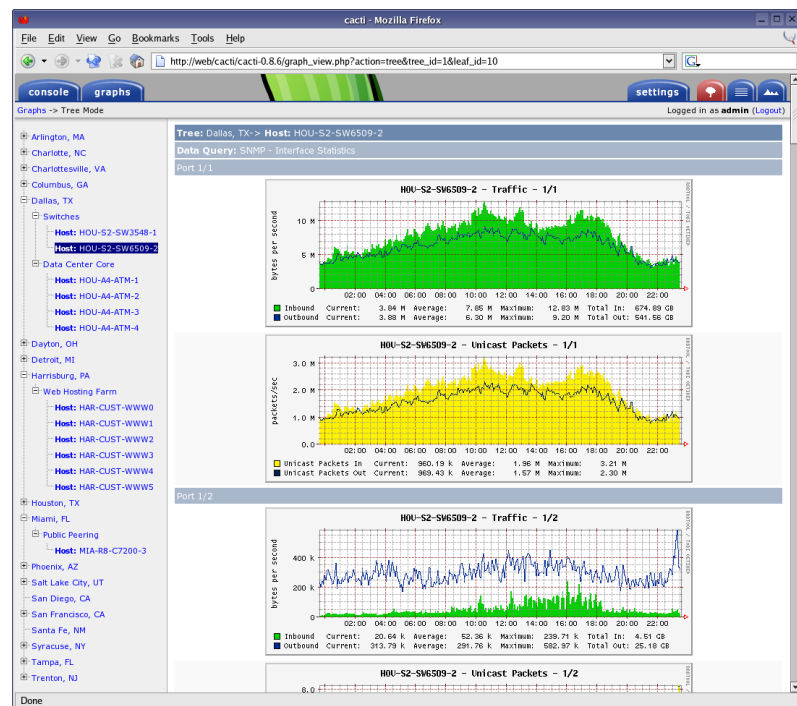
Seurattavan tiedon noutaminen rakentuu kahdesta osasta. Ensimmäinen on Cactille kerrottu laite (Device). Laitetiedot sisältävät laitteen sijainnin ja laitekohtaiset SNMP asetukset. Toisena asetetaan datamalli, joka osoittaa tarkemmin mistä noudettava tieto löytyy ja kuinka data tallennetaan (RRD-tietokannan nimi ja RRA-arkistojen rakenne). Näistä syntyy Cactille luettava datalähde (Data Source), joiden arvoja piirretään kuvioihin. (Berry ym. 2013.)

2.8.3. Datan esittäminen

Monitorointi ohjelmana Cactin tärkein osa on kerätyistä arvoista piirretyt kuviot ja näiden mahdollistama muutoksien seuranta. Kuviot generoidaan käyttäen RRDtoolin piirto-ominaisuuksia. Cacti tukee suurinta osaa RRDtoolin eri asetuksia, viivojen asetuksista – arvojen käsittelyyn ennen piirtämistä. (Berry ym. 2013.)

Kuvioita voidaan katsoa kolmessa eri moodissa:

- *List Mode*: Yksinkertainen listaus, jossa kuvioita ei piirretä vaan jokaisen kuvion nimi, lähdelaitte, kuvio-malli ja koko tiedot listataan.
- *Preview Mode*: Kaikki kuviot piirretään pieninä esikatselu ikkunoina kahdelle sarakkeelle.
- *Tree View*: Cactin päänäkö, kuviot piirretään täydessä koossa. Näkymän vasemmassa reunassa on puumainen kansiorakenne, jonka rakennetta ja sisältöä voidaan muokata käyttäjän toimesta. Tree View näkymä kuviossa 12. (Berry ym. 2013.)



Kuvio 12 Tree View Cactissa (Screen Shots n.d.)

2.8.4. Liitännäiset

Cactissa on rajapinta, joka sallii Cactin toiminnan muokkaamista ja laajentamista käyttäjän toimesta. Rajapinta on nimeltään *Plugin Architecture* ja antaa nimen mukaisesti Cactille tuen liitännäisille. Cactille on olemassa useita eri liitännäisiä ja laajennuksia, joita ylläpitävät eri henkilöt tai yhteisöt. Liitännäiset tai laajennukset

asetetaan Cactin pääkansioista löytyvään *Plugins*-kansioon. Liitännäisten ja laajennusten lopullinen asentaminen ja käynnistäminen tapahtuvat Cactin käyttöliittymässä. Kuviossa 13 on liitännäisten hallintanäkymä. (Installing the Plugin Architecture n.d.)

Plugin Management (Cacti Version: 0.8.7g, Plugin Architecture Version: 2.8)

Search:

Go

Clear

Showing All 6 Rows

Actions	Name	Load Order**	Description	Type	Status	Author	Web Page	Version
	Settings		Global Plugin Settings	System	Active	Jimmy Conner	http://cactiusers.org	0.6
	Autom8		Automate Cacti Tasks	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin/autom8	0.3506
	Nectar		Send Graphs via Email	General	Installed	Reinhard Scheck	http://docs.cacti.net/plugin/nectar	0.286
	Aggregate		Create Aggregate Graphs	General	Active	Reinhard Scheck	http://docs.cacti.net/plugin/aggregate	0.0795
	Discovery		Network Discovery	General	Active	Jimmy Conner	http://cactiusers.org	1.1
	Rrdclean		RRD File Cleaner	General	Not Installed	Reinhard Scheck	http://docs.cacti.net/plugin/rrdclean	0.40

Showing All 6 Rows

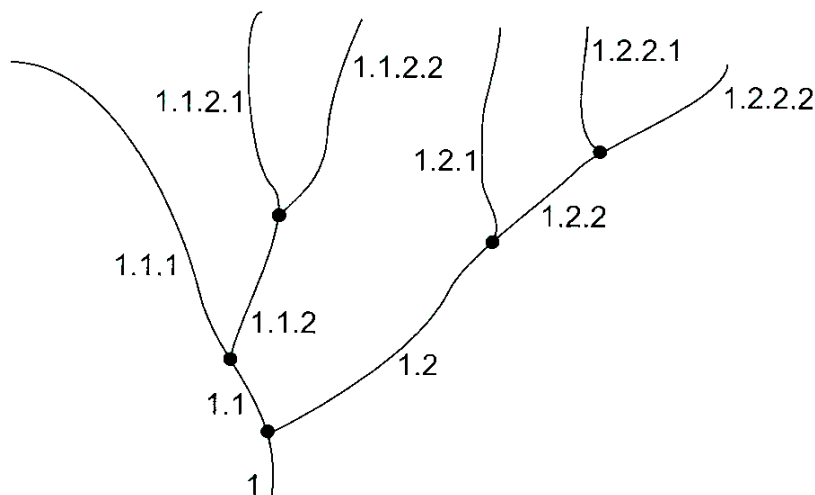
NOTE: Please sort by 'Load Order' to change plugin load ordering.

NOTE: SYSTEM plugins can not be ordered.

Kuvio 13 Cactin liitännäisten hallintanäkymä (Installing the Plugin Architecture n.d.)

2.9. Yksilöintitunniste

Yksilöintitunniste (OID, Object Identifier) on ITU:n (International Telecommunication Union) ja ISO:n (International Organization for Standardization) kehittämä järjestelmä, jolla voidaan eritellä mikä tahansa objekti, konsepti tai ”asia” maailmanlaajuisesti yksiselitteisellä tunnisteella. OID rakentuu kokonaisluvuista ja pisteistä, jolla eritellään yksittäinen kohde hierarkkisessa nimiavaruudessa, kts. kuvio 14. (OID Repository n.d.)



Kuvio 14 Yksilöintitunniste puu (Kimmo Kaario 2002)

Yksilöintitunnuksen muodostamisvastuu rakentuu hajakeskistetyksi ylhäältä alas, kunkin alipuun haltija vastaa koko oman alipuun asianmukaisesta käytöstä ja hallinnoinnista. Esimerkiksi, ISO hallitsee OID puuta 1. ISO on varannut tästä puusta alipuun 2 ISO:n jäsenrungoksi. Tästä jäsenrungosta ISO on osoittanut Suomelle OID:n 246, jonka hallitsijaksi on osoitettu Suomen Standardisoimisliitto. (OID Repository n.d.)

1 - International Organization for Standardizationin hallitsema OID puu.

1.2 - ISO:n jäsenrunko

1.2.246 - Suomi – Suomen Standardisoimisliiton hallitsema alipuu.

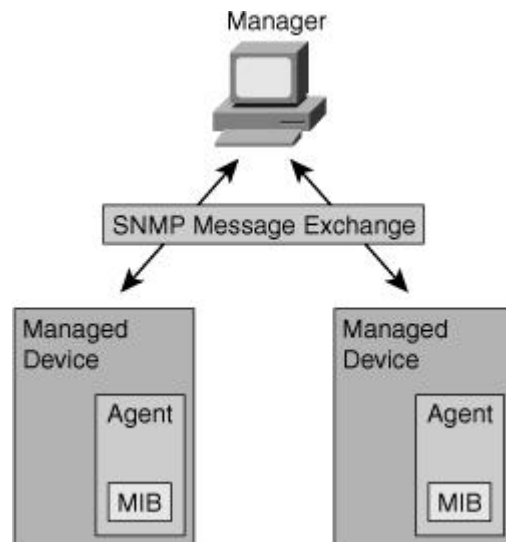
2.10. SNMP

SNMP (Simple Network Management Protocol) on TCP/IP-protokollaperheeseen (Transmission Control Protocol / Internet Protocol) kuuluva verkonhallinta-protokolla, jonka ensimmäisen version IETF (Internet Engineering Task Force) määritteli vuonna 1988. Tarkoituksena saada aikaiseksi yksinkertainen, hyvin vähäisiä resursseja sekä vähän tiedonsiirto kaistaa käyttävä protokolla, jolla hallita räjähdysmäisesti kasvavia TCP/IP-verkkoja. SNMP:n käyttö kasvoi nopeasti ja on tänäkin päivänä laajassa käytössä yksinkertaisen toimintansa vuoksi. (Introduction to SNMP 2005.)

Protokollasta on määritelty kolme eri versiota, SNMPv1, SNMPv2 ja SNMPv3. Versio 2 laajensi SNMP:n toimintaa sekä lisäsi yksinkertaisia tietoturva-asetuksia. Versiossa 3 tietoturvaa laajennettiin edelleen, lisäten käyttäjän todennuksen sekä salauksen. (Kimmo Kaario 2002, 270.)

SNMP määrittää palvelin-asiakas yhteyden. Asiakas toimii SNMP verkon hallitsijana ja hallittavat laitteet palvelimina, kts. Kuvio 15. Asiakasta kutsutaan hallinta-asemaksi (SNMP Manager), joka lähettää kyselyitä sekä käskyjä hallinta-agenteiksi kutsuttaville palvelimille (SNMP Agent). Hallinta-agentit ohjaavat seurattavia laitteita ja asettavat laitteiden tietoja hallinta-aseman saataville. Tiedonsiirto SNMP osapuolien välillä

tapahtuu UDP-protokollalla (User Datagram Protocol). Näiden lisäksi SNMP:lle on määritelty hallittavien kohteiden hallintatietokanta, (MIB, Management Information Base) sekä tieto-olioiden (Objekti) rakenne (SMI, Structure of Management Information). (Kimmo Kaario 2002, 270.)



Kuvio 15 SNMP verkko (Hoda 2005.)

Hallittavat tiedot näkyy SNMP:ssä joukkona objekteja, jotka sijaitsevat MIB:ssä. Eri objekteja tai näiden joukkoja määritetään MIB-moduuleissa. SMI määrittää miten objektit kuvataan MIB:ssä. SMI standartissa määritellään kehyksiä kolmelle eri osalle. (RFC 2578.)

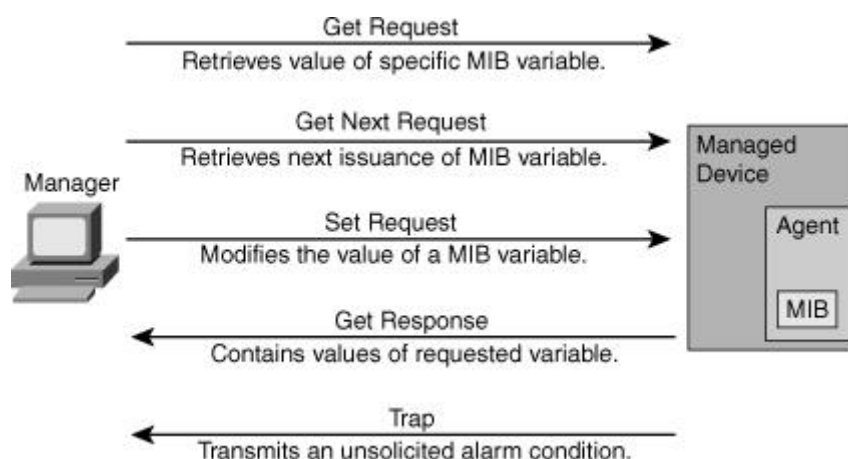
- *Moduulin määrittäminen:* Kuvataan moduulin tietoja. Osoitetaan muun muassa moduulin kohta nimiavaruudessa, moduulin ylläpitäjä ja kuvataan moduulin sisältöä sekä yhteyksiä muihin moduuleihin.
- *Objektien määrittäminen:* Osoitetaan luettava ja/tai asetettava kohde. Objektilla määritellään muun muassa Objektin yksilöintitunnus, nimi, jokin lyhyt kuvaus ja minkä tyyppisen arvon tai tiedon se pitää sisällään.
- *Ilmoituksien määrittäminen:* Kuvataan hallinta-agentin itsenäisesti lähetettäviä viestejä. Ilmoitukselle määritellään monia samoja tietoja kuin objektin määrittämisessä. (RFC 2578.)

MIB rakentuu hallinta-agentille asennetuista MIB-moduuleista. MIB-moduulit sisältävät eri kokonaisuuksia ja rakentuvat SMI:ssä määritetyillä kehyksillä. MIB-moduuleja ylläpitävät eri tahot. Verkkolaitteiden valmistajat esimerkiksi ylläpitävät moduuleja, jotka erittelevät heidän laitteiden uniikkeja objekteja. (Introduction to SNMP 2005.)

Hallittavalla laitteella oleva hallinta-agentti ylläpitää hallintatietokannassa määritettyjen kohteiden tietoja ja arvoja. Agentti lähettää hallintatietokannan kohteiden tietoja kyselevälle hallinta-asemalle. (Introduction to SNMP 2005.)

2.10.1. Viestit

SNMP:llä on viisi viestityyppiä. Hallinta-asema voi lukea tai kirjoittaa hallinta-agentilta löytyvien kohteiden arvoja. Hallinta-aseman on myös mahdollista kulkea läpi hallinta-agentin hallintatietokannan rakenne tai sen osia. Hallinta-agentilla on mahdollista lähettää ilmoitusviesti hallinta-asemalle. Hallittavat kohteet eritellään viesteissä yksilöintitunnuksella. Viestien kulkusuunnat ovat nähtävissä kuviossa 16. (Kimmo Kaario 2002, 277–279.)



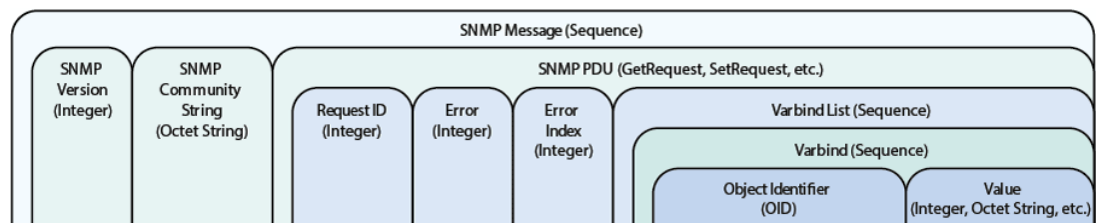
Kuvio 16 SNMP:n viestit (Hoda 2005.)

- **GetRequest:** Hallinta-asema pyytää hallinta-agentilta yksittäisen kohteen arvon tai tiedon. SNMPv2 lisäsi myös **GetBulkRequest** kyselyn, jolla hallinta-agentilta pyydetään useiden kohteiden tietoja yhdellä viestillä.

- **GetNextRequest:** hallinta-asema kysyy hallinta-agentilta jonkin kohteen seuraavan kohteen tiedot. Hallinta-asema voi tiedustella hallinta-agentin hallintatietokannan rakenteen lähettämällä perättäisiä **GetNext** kyselyjä, tämä tunnetaan useissa hallinta-agenteissa komentona **Walk**.
- **SetRequest:** Hallinta-asema kirjoittaa hallinta-agentille jonkin kohteen arvon tai asetuksen.
- **GetResponse:** Vastausviesti hallinta-agentilta hallinta-asemalle.
- **Trap:** Hallinta-agentin itsenäisesti hallinta-asemalle lähettämä ilmoitusviesti. Käytetään usein hälytys- tai varoitusviestinä. (Kimmo Kaario 2002, 277–279.)

2.10.2. SNMP viesti kehys

SNMP-viestikehys rakenteen näkyy kuviossa 17. Kaikki SNMP viestit sisältävät SNMP Version, SNMP Community String sekä SNMP PDU -kentän. PDU-kentän tarkka rakenne riippuu viestistä. Kenttien selitykset ovat kirjoitettuna alla. (Bruey 2005.)



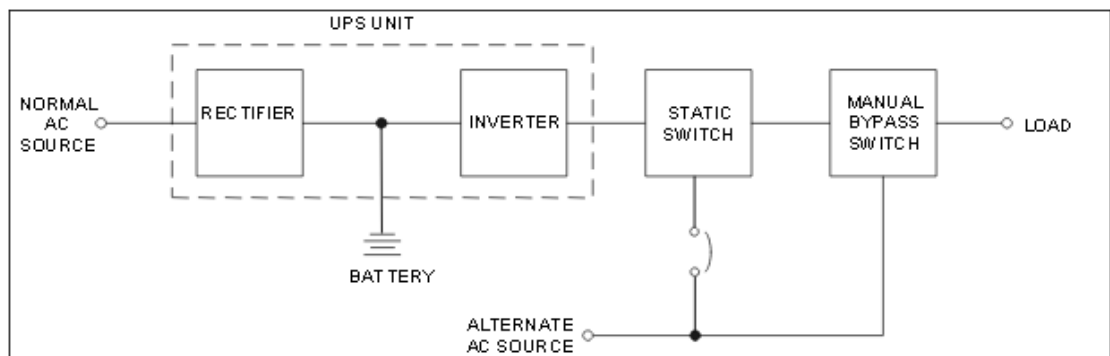
Kuvio 17 SNMP kehys (Bruey 2005)

- **SNMP Version:** Kokonaisluku, joka kertoo SNMP version.
- **SNMP Community String:** Salasana merkkijono.
- **Request ID:** Kokonaisluku, joka kertoo viestin tyyppin.
- **Error:** Kertoo hallinta-asemalle, jos hallinta-agentti kohtasi virheen.
- **Error Index:** Kertoo hallinta-asemalle, mikä yksilöintitunnus aiheutti virheen.
- **Object Identifier:** Yksilöintitunnus, joka kertoo haettavan objektin.
- **Value:** Noudetun kohteen arvo/tieto tai uusi kohteeseen asetettava arvo/asetus. (Bruey 2005.)

2.11. Keskeytymätön virransyöttö

Keskeytymätöntä virransyöttöä (UPS, Uninterruptible Power Supply) käytetään ylläpitämään ja suojaamaan tärkeitä tai herkkiä laitteita jännitekatkoilta tai häiritseviltä jännitemuutoksilta. UPS:n perusrakenteen osat on esitetty kuviossa 18. Perusosia on kolme: tasasuuntaaja (rectifier), akusto (battery) ja invertteri. Näiden lisäksi UPS:ssa voi olla muita jännitelähteitä (alternate AC source) tai ohituskytkin (bypass switch), jos UPS täytyy huoltaa sammuttamatta ylläpidettäviä laitteita (load). (TM 5-693 2002.)

UPS asetetaan ylläpidettävän laitteen ja jännitelähteen väliin. Tasasuuntaaja lataa akustoa sekä invertteri syöttää ylläpidettäville laitteille toimintajännitteen. Jos UPS:in tulojännite katkeaa, ylläpidettävät laitteet siirretään akkujen ylläpidettäväksi katkon ajaksi. (TM 5-693 2002.)



Kuvio 18 UPS järjestelmän kaavio (TM 5-693 2002)

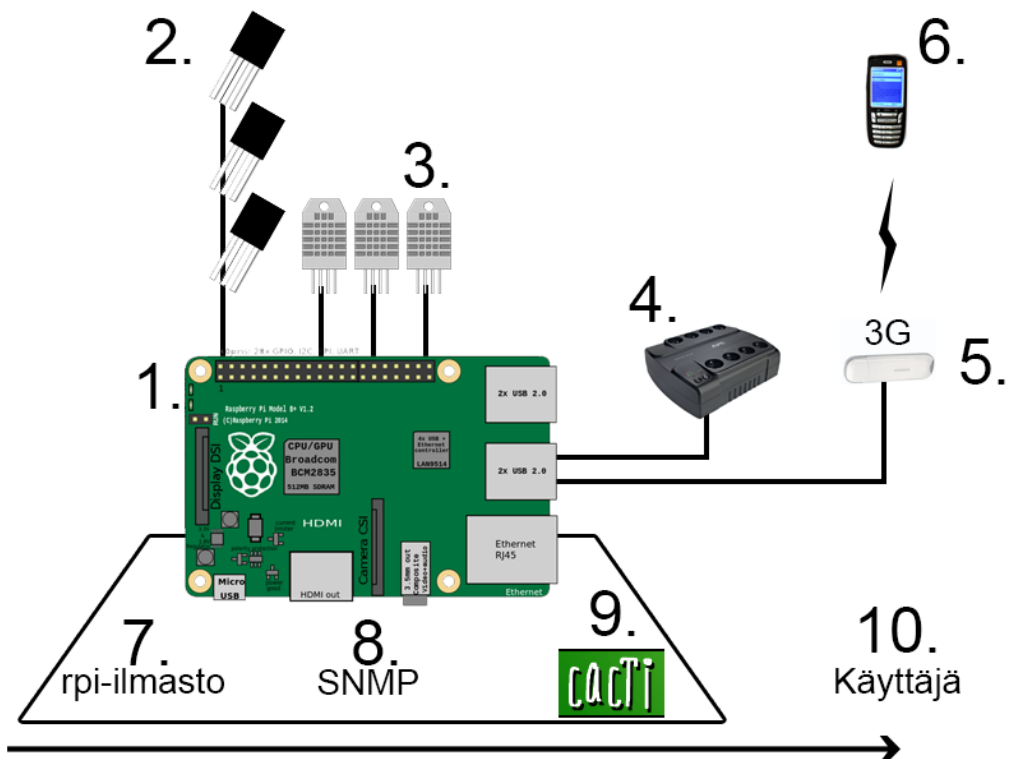
3. Järjestelmän kuvaus

Järjestelmä on toteutettu Raspberry Pi 2 -pientietokoneelle. Järjestelmän toiminta rakentuu kahdesta osa-alueesta: etäkyselystä ja valvonnasta. Etäkysely rakentuu laitetilaa seuraavista antureista ja ohjelmistoista, joilla ne saadaan etäluettavaksi SNMP:n avulla. Valvonta rakentuu *Cactista*, joka on WWW-selaimella käytettävä valvontaohjelmisto, jota käytetään antureista saatujen arvojen keräämiseen sekä niistä luettavien muutos käyrien piirtämiseen. Toimeksiantajan toiveesta *Cactin* toimintaa laajennettiin lisäämällä siihen mahdollisuus asettaa hälytysraja-arvoja sekä saamaan näistä tapahtumista tekstiviestihälytyksiä.

Kuviossa 19 on nähtävissä järjestelmän eri osat:

1. Raspberry Pi 2.
2. Lämpötilaa mittaavat DS18B20-anturit ovat liitettynä GPIO-väylään. Anturit käyttävät 1-Wire protokollaa kommunikoidessaan Raspberry Pi:n kanssa. Kaikki DS18B20 anturit liitetään toisiinsa joko sarjaan tai rinnakkain.
3. Lämpötilaa ja kosteutta mittaavat AM2302-anturit. AM2302-anturit ovat myös liitettynä GPIO-väylään. Jokainen AM2302-anturin DATA-pinni vaatii itselleen oman GPIO:n.
4. American Power Conversion Corporationin (APC) valmistama keskeytymätön virransyöttö, joka on liitettynä USB-porttiin. Raspberry Pi kerää käyttäjää kiinnostavia tietoja UPS:in tämänhetkisestä tilasta.
5. 3G-modeemi, joka on kytkettynä USB-porttiin. Järjestelmä käyttää 3G-modeemia lähettäessään tekstiviestejä valittuihin numeroihin, jos käyttäjän asettaman raja-arvot ylitetään.
6. Käyttäjän puhelin, joka vastaanottaa hälytyksiä.
7. Ohjelmisto, joka luotiin keräämään arvot antureilta sekä UPS:lta. Kyselee antureiden sekä UPS:n tiedot puolen minuutin välein, asettaen ne lyhyen ajan tietokantaan.

8. SNMP saadaan järjestelmälle Net-SNMP ohjelmistopakettilla. Net-SNMP:n hallinta-agenttia laajennetaan, jolloin *rpi-ilmaston* keräävät arvot saadaan käyttäjän luettavaksi etänä. Valvontaa varten *Cacti* kerää arvot omaan pysyvään tietokantaansa käyttäen paikallisia SNMP kyselyitä.
9. *Cacti* kerää niin antureilta ja UPS:lta arvoja tietokantoihinsa, kuin myös Raspberry Pi:n perustilanne arvoja. Piirtää arvoista käyttäjälle käyriä, joista käyttäjä voi katsoa seurattavien kohteiden tämänhetkisiä arvoja sekä muutoksia. Käyttäjä voi myös tarvittaessa asettaa seurattaville kohteille raja-arvoja, joiden ylittyessä käyttäjälle lähetetään tekstiviesti hälytys.
10. Käyttäjälle kohteiden seuranta mahdollistetaan *Cactilla*, joka on saatavilla WWW-selaimella. Käyttäjä voi WWW-selaimesta asettaa *Cactille* seurattavia kohteita, raja-arvoja ja puhelinnumeroita. Käyttäjä voi myös muokata kuinka ja miltä ajanjaksolta kuviot piirretään ja esitetään.



Kuvio 19 Järjestelmän osat

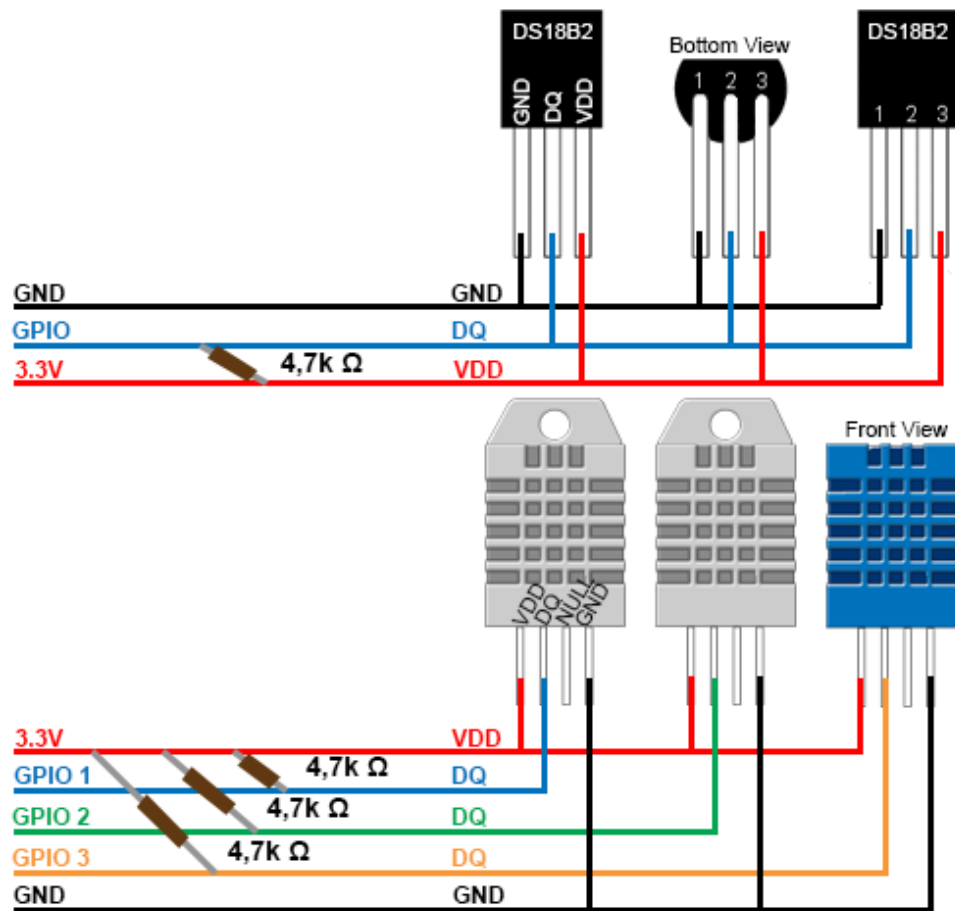
3.1. Raspberry Pi 2

Käyttöjärjestelmänä toimii Raspberry Pi:tä varten kehitetty Rasbian Linux, joka on Debian Linuxista haaroitettu käyttöjärjestelmä. Käyttöjärjestelmän uusin versio kirjoitushetkellä pohjautuu Debianin versioon Jessie. Raspberry Pi:lle asennettiin NET-SNMP ohjelmistopaketti. NET-SNMP ohjelmistopaketti sisältää ison määrän SNMP:tä käyttäviä ja tukevia ohjelmia, joista tärkein on SNMP kyselyitä vastaanottava *snmpd* hallinta-agentti ohjelma. Tämä mahdollistaa antureiden sekä erillisten Raspberry Pi:n tietojen ja arvojen etäkyselyn SNMP protokollalla.

3.2. Anturit

Koska Raspberry Pi:n GPIO-väylä on digitaalinen, olisi analogisten komponenttien käyttäminen vaatinut erillisen analogia-digitaalimuuntimen hankinnan. Analogisia antureita ei siis lyhyen pohdinnan jälkeen työhön oikeastaan edes harkittu. Digitaalisten antureiden hinnoissa ei ollut kovinkaan paljon eroja, joten työhön valittiin AM2302- ja DS18B20-anturit, joiden käytöstä Raspberry Pi:llä löydettiin huomattavasti dokumentointia. AM2302-anturilla voidaan mitata ilman kosteutta ja lämpötilaa. DS18B20 on 1-Wire protokollalla kommunikoiva lämpötila-anturi.

Anturit liitetään Raspberry Pi:n GPIO-väylään. DS18B20-anturit rakentuvat kolmesta pinnistä: anturin toimintavirta-, data- ja maa-pinnistä. DS18B20-anturit voidaan asentaa kokonaisuudessaan sarjaan ja ympäri seurattavaa tilaa. AM2302-anturi vaatii toimiakseen samaan tapaan myös kolme pinniä. Antureiden pinnien liittäminen periaate on kuviossa 20. AM2302-anturit käyttävät hiukan 1-Wireä muistuttavaa protokollaa, joka ei tue useita antureita yhden data-pinnin varassa. Täten jokaiselle AM2302-anturille täytyy antaa oma GPIO sen data-pinnille, vaikka ne muuten jakavat GPIO-väylän virta- ja maa-pinnien osalta. Data-väylät täytyy saada lepotilassa ylös, joten ne liitetään virta-pinniin 4,7 kilo ohmin vastuksella.

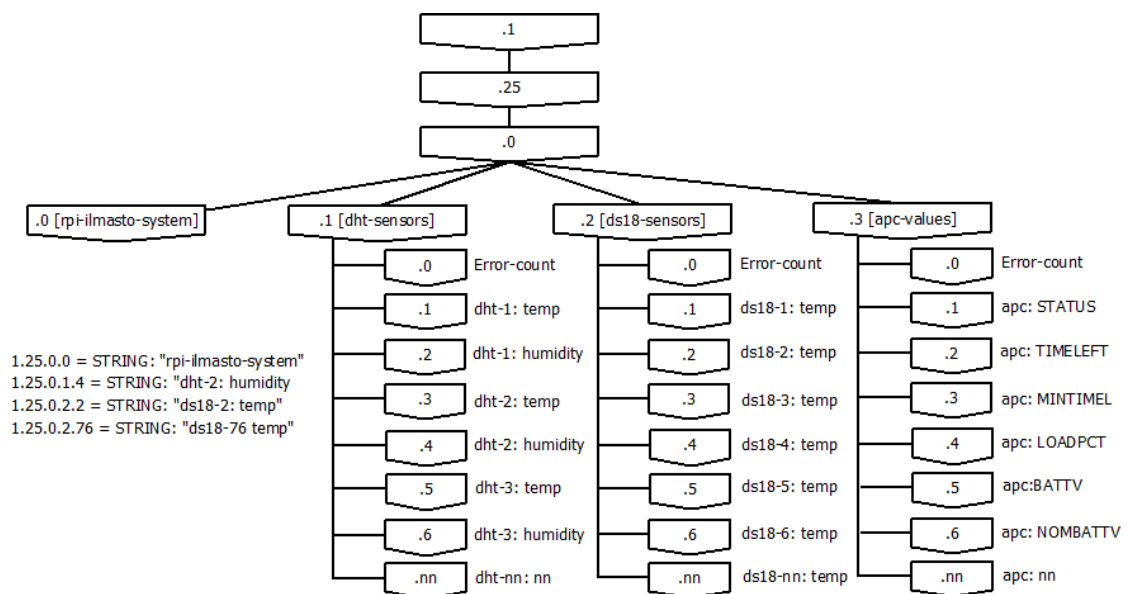


Kuvio 20 Antureiden pinnien liitântä sekä Raspberry Pi:n GPIO-väylä

4. rpi-ilmasto

Jotta antureiden ja muiden seurattavien kohteiden arvot saadaan saataville SNMP:llä, järjestelmää varten luotiin joukko bash-komentotulkillla ajettavia skriptejä. Skripteistä ja muista tiedostoista rakentuva kokonaisuus nimettiin *rpi-ilmastoksi*. *Rpi-ilmasto* ylläpitää jokaiselle seurattavalle anturille tai kohteelle 24 tunnin pituista round robin -tietokantaa, joista käyttäjän on mahdollista hakea antureiden tai kohteiden viimeisin arvo käyttäen SNMP:tä. *Rpi-ilmasto* koostuu seitsemästä skriptistä, konfigurointitiedostosta, tietokannoista, aputiedostosta sekä skriptien luomista loki tiedostoista.

Rpi-ilmasto mahdollistaa käyttäjää osoittamaan sille seurattavia kohteita, joista skriptit keräävät arvoja ja asettavat ne tietokantaan. *Rpi-ilmasto* rakentaa automaattisesti konfiguroiduista kohteista OID puun, jota käyttäjä voi kysellä SNMP:n GET hakua käyttäen. Kuviossa 21 on nähtävissä esimerkki, miten puu lähtee rakentumaan. Kuviossa käyttäjä on valinnut root OID:ksi 1.25.0, joten *rpi-ilmaston* oksat rakennetaan sen jälkeen. Yksi oksa jatkuu niin pitkälle kuin käyttäjä on asettanut antureita konfiguraatiotiedostoon.



Kuvio 21 *Rpi-ilmaston* OID puu

Jos skriptit ja riippuvuudet on asennettu, anturit liitetty Raspberry Pi:hin, niiden vaatimat protokollat aktivoitu sekä *snmpd* konfiguroitu ja käynnistetty, voidaan *rpi.cfg*:hen lisätä seurattavat kohteet. Kun skriptit käynnistettään komennolla "*rpi-ilmasto start*", voidaan antureiden arvoja noutaa SNMP:llä.

Kuviossa 22 on *rpi-ilmasto*a koestettaessa luotu alipuu kuljettuna. Kuviosta nähdään, että *rpi-ilmasto*lle on konfiguroitu yksi DHT-anturi virheellisesti (21 tuhatta virhettä), kolme DS18B20-anturia sekä muistikortin aktiivisuuden seuraaminen. Keskeytymättömän virransyötön seuranta ei ole käynnistetty, joten *snmpd* antaa ainoastaan *apc-ilmaston* virhelaskurin arvon, joka näyttää nollaa.

```
RPIILMASTO-MIB::rpi-ilmasto-null = STRING: "rpi-ilmasto system"
RPIILMASTO-MIB::dht-sensors = STRING: "dht-ilmasto sensors"
RPIILMASTO-MIB::dht-errors = STRING: "21770"
RPIILMASTO-MIB::dht-1-temperature = STRING: "nothing to read dht-1.rrd"
RPIILMASTO-MIB::dht-1-humidity = STRING: "nothing to read dht-1.rrd"
RPIILMASTO-MIB::ds18-sensors = STRING: "ds18-ilmasto sensors"
RPIILMASTO-MIB::ds18-errors = STRING: "0"
RPIILMASTO-MIB::ds18-1 = STRING: "25.312"
RPIILMASTO-MIB::ds18-2 = STRING: "25.5"
RPIILMASTO-MIB::ds18-3 = STRING: "25.375"
RPIILMASTO-MIB::apc-information = STRING: "apc-ilmasto values"
RPIILMASTO-MIB::apc-errors = STRING: "0"
RPIILMASTO-MIB::disk-information = STRING: "system disk stats"
RPIILMASTO-MIB::disk-iops = STRING: "0.49"
RPIILMASTO-MIB::disk-reading = STRING: "2.01"
RPIILMASTO-MIB::disk-writing = STRING: "3.96"
RPIILMASTO-MIB::disk-read = STRING: "169615"
RPIILMASTO-MIB::disk-written = STRING: "333548"
```

Kuvio 22 Rpi-ilmaston kulkeminen SNMP:llä

4.1. Tiedostot & riippuvuudet

Toimiakseen *rpi-ilmasto* tarvitsee Linux-käyttöjärjestelmän lisäksi:

- **python**: Käytetään ajamaan Adafruitin DHT -kirjastolla AM2302-antureita.
- **Adafruit DHT -kirjasto**.
- **rrdtool**: Tietokanta työkalu, jota käytetään antureista saatujen arvojen tallentamiseen.
- **snmpd**: SNMP-agentti ohjelma, joka sallii antureista saatujen arvojen kyselyn SNMP:llä.
- **iostat**: Ohjelma, joka seuraa muistikortin aktiivisuutta, osa sysstat pakettia.
- **apcupsd**: APC:n tuottamien UPS-laitteiden ohjausohjelmisto.

Ohjelmisto vaatii, että pääkansio on asennettu kohteeseen */usr/local/rpi-ilmasto*.
Tiedostorakenne:

```

.rpi-ilmasto/
|
+----- .bin/
|           +----- rpi-ilmasto
|           +----- dht-ilmasto
|           +----- ds18-ilmasto
|           +----- apc-ilmasto
|           +----- iostat-ilmasto
|           +----- null-ilmasto
|           +----- simple
|           `----- rpi.cfg
|
+----- .log/
|           `----- *.log
|
+----- .databases/
|           +----- *-0.rrd
|           +----- dht-*.rrd
|           +----- ds18-*.rrd
|           +----- apc-*.rrd
|           `----- io.rrd
|
`----- .doc/
           `----- rpi-ilmasto.man

```

- Pääkansio

- Skriptien sijainti

- Lokitiedostot

- Tietokannat

- Dokumentit

4.2. Skriptit

4.2.1. INSTALL

INSTALL on asennus skripti. *INSTALL* ei ole osa *Rpi-ilmasto*a, vaan on luotu helpottamaan sen asentamista ja antureiden käyttöönottoa. Kuviossa 23 on nähtävissä asennus skriptin *help* tuloste. Skripti asentaa antureiden vaatimat riippuvuudet, siirtää *rpi-ilmaston* oikeaan kohteeseen, tekee symbolisen linkin, jotta *rpi-ilmasto*a voidaan käyttää mistä tahansa sekä tekee muutoksia eri ohjelmien konfiguraatioihin automaattisesti. Skripti kysyy asentajalta minkä OID:n hän haluaa valita, mahdollistaa oletusasetusten asettamisen *snmpd*:lle, sekä lisää suositeltuja konfigurointeja *crontabille* ja *logrotatelle*.

```
@kotiraspijeah ~/ $ sudo ./INSTALL -h

This is simple and ugly script to
install rpi-ilmasto automatically.
It will check for dependencies and will
install any missing ones and
configures crontab and logrotate for you
Installer will also overwrite snmpd.conf
with default settings for rpi-ilmasto
if told to do so

-----
Default snmpd settings:

snmpd port: 161
v2c user : rpiuser
disk / 100000
sysServices 72

default will limit reading to following OIDs:
view your OID root
view 1.3.6.1.4.1.2021.4 memory information
view 1.3.6.1.4.1.2021.9.1 disk status
view 1.3.6.1.4.1.2021.10.1.3 load status
view 1.3.6.1.2.1.1.3.0 uptime

-----
Options:
  -h      Read this help text.
  -c      Check system for installation
          and dependencies
  -s      Run the installer silently,
          only asking for OID and
          1-wire settings. Will
          overwrite your snmpd
          settings WITHOUT prompt

-----
```

Kuvio 23 *INSTALL*-skriptin *help* tuloste

4.2.2. Ohjaus

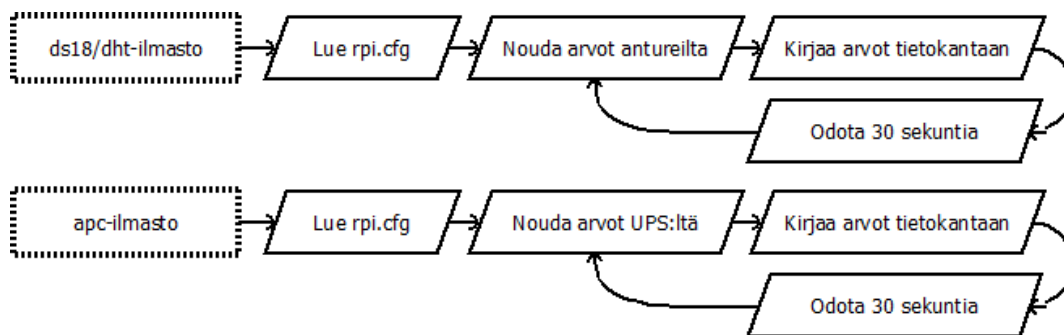
Kerääjäskriptejä ohjataan *rpi-ilmasto* -skriptillä, joka riippuen sille annetuista muuttujista käynnistää (*start*), sammuttaa (*stop*) tai kertoo kerääjäskriptien tilan (*status*). Skriptillä voidaan avata muokattavaksi myös konfiguraatiotiedosto (*edit*) tai lukea englanninkielinen aputiedosto (*help*). Kuviossa 24 on viesti, jonka skripti antaa, jos se ei saa muuttujaa.

```
@kotiraspijeah ~ $ sudo rpi-ilmasto
[NOTICE] Usage: service rpi-ilmasto {start|stop|status|edit|help}
```

Kuvio 24 *rpi-ilmaston* vihje viesti

4.2.3. Kerääjät

Neljä skriptiä toimivat kyseltävien kohteiden arvojen kerääjinä. Nämä ovat *dht-ilmasto*, *ds18-ilmasto*, *apc-ilmasto* ja *iostat-ilmasto*. Kerääjäskriptit lukevat *rpi.cfg* konfiguraatiotiedostosta haettavat kohteet ja hakevat näistä kohteista kerättävät arvot, jotka kirjataan käyttäjän numerolla osoittamaan tietokantaan. Antureiden määrää ei ole rajoitettu. Kuviossa 25 on hyvin yksinkertaistettu vuokaavio kerääjäskriptien toiminnasta.



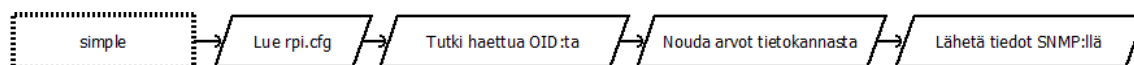
Kuvio 25 Muutaman kerääjäskriptin toiminta

Tietokannat luodaan käyttäen RRDtool-ohjelmaa. Skriptit luovat jokaiselle asetetulle anturille tai kohteelle tietokannan automaattisesti. Tietokannassa on arvoja jokaiselle minuutille viimeiseltä 24 tunnilta, joten käyttäjän on mahdollista noutaa aikaisempia arvoja käyttäen komentoriviä ja *rrdtool*-sovellusta. Kerääjäskriptit luovat

lokitiedostoja, joihin merkataan käynnistys/sammutus sekä mahdolliset virheet tai varoitukset, joita skriptit kohtaavat.

4.2.4. SNMP laajennus

Net-SNMP:n agentin toimintaa laajennetaan *simple* nimisellä skriptillä. Käyttäjä valitsee itselleen OID-juuren, johon osoitetut SNMP kyselyt lähetetään *simple*-skriptille. Skripti lukee *rpi.cfg* konfiguraatiotiedostosta käyttäjän valitseman OID:n ja luo siihen viisi oksaa. Yksi oksa on *.0*, joka lähettää vastaukseksi aina viestin "*rpi-ilmasto-system*". Oksa *.1* on DHT-antureille, jonka alle skripti asettaa käyttäjän merkitsemät DHT-anturit. Oksa *.2* on DS18B20-antureille, *.3* on keskeytymättömän virransyötön arvoille sekä *.4* on iostatin antamille muistikortin aktiivisuusarvoille. *Simple*-skripti tutkii sille lähetettyä OID:ta ja pääättelee mitä kohdetta kysellään ja onko kohdetta olemassa. Kuviossa 26 on yksinkertaistettu vuokaavio skriptin toiminnasta.



Kuvio 26 SNMP laajennus-skriptin toiminta

4.2.5. Nollaaja

Nollaimella antureiden tietokantojen arvot voidaan asettaa tuntemattomiksi tai virhelaskureiden arvot nollassi. *Null-ilmaston* voi ajaa manuaalisesti, mutta suositellaan ajettavaksi esimerkiksi *crontabilla* muutaman kerran päivässä. Nollaimelle voi antaa kaksi eri muuntajaa. Jos annettu muuntaja on "*0*", nollain muuttaa virhelaskureiden arvot nolliksi. Jos annettu muuttuja on "*rest*", nollain asettaa tietokantojen arvot tuntemattomiksi.

4.3. Konfiguraatietiedosto

Ohjelmiston skripteille osoitetaan kohteet sekä root OID käyttäen *rpi.cfg*-konfiguraatietiedostoa. Konfiguraatietiedostoon voi laittaa seuraavia asetuksia:

[oid \$ROOT]	- Osoittaa OID kohdan, johon rpi-ilmaston alipuu rakennetaan ja mihin snmpd vastaa.
[\$smalli \$GPIO \$dnro]	- Osoittaa DHT tyyppisen anturin GPIO:n mukaan sekä valitsee sille OID:n.
[ds18 \$hwid \$dnro]	- Osoittaa DS18B20 anturin rautatunnisteella sekä valitsee sille OID:n.
[apc on]	- Sallii UPS kysely skriptin käynnistystä ja mahdollistaa sen tilan kyselyn SNMP:llä.
[disk on]	- Sallii muistikortin IO-tapahtumien seurannan SNMP:llä.

4.4. Muut tiedostot

Skriptien mukaan on kirjoitettu englanniksi manuaalisivu (manpage). Sivulle on kirjoitettu läpi skriptien toiminta, sijainnit, vaaditut riippuvuudet sekä ohjeet skriptien käyttöönottoon, mukaan lukien *snmpd*:n ja *Logrotaten* konfigurointi. Kuviossa 27 on näkyvissä pieni ote manuaalisivua. Manuaalisivu on kokonaisuudessaan liitteenä 1.

```

CONFIGURATION
Options
    oid OID

    Used to choose the root oid for the SNMP protocol to query and
    respond to. OID must begin with a period [.] and end to a numä
    ber.

MODEL GPIO NUMBER

This setting is for DHT/AM type tempeture/humidity sensors,
MODEL is 11, 22 or 2302 [for the models DHT-11, DHT-22 or
AM2302]. GPIO is the General Purpose I/O to which the sensors
DATA pin has been connected to. NUMBER is what you wish to
define as the identification for the sensor. Also acts as the
last digit of the OID that can be used to query this sensor with
SNMP. DHT sensors have OID .1.n.

```

Kuvio 27 Ote rpi-ilmaston manpagesta.

Luotua OID-alipuuta varten on luotu MIB-moduulin malli tiedosto RPIILMASTO-MIB.template. Tiedostoon on määritetty kummallekin anturille 20 kohdetta sekä *apcupsd*:n ja *iostat*:n tulosteet. MIB-moduuli on luettavissa liitteenä 2.

5. Etäkysely

Etäkyselyllä Raspberry Pi:hin liitettyjen antureiden ja laitteiden arvot saadaan käyttäjän luettavaksi etänä. Raspberry Pi:n GPIO-pinneihin on liitetty kolme AM2302-anturia sekä USB-porttiin APC:n valmistama UPS. Antureilta sekä UPS:ilta kerätään arvoja tietokantoihin, joista voidaan SNMP:n avulla noutaa viimeisimmät arvot etänä.

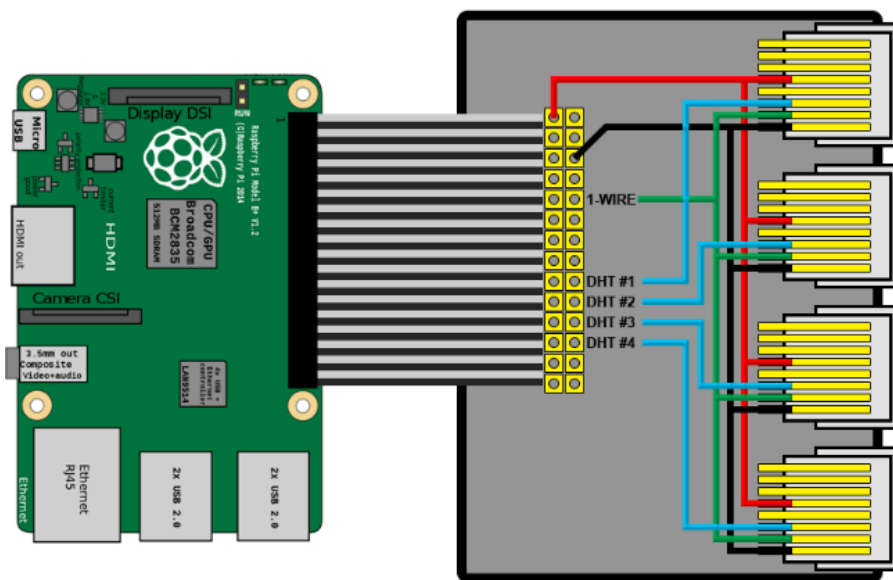
5.1. Antureiden kytkentä

Antureita varten toimeksiantajan kanssa suunniteltiin liitäntä tapa, jolla antureiden käyttöä voidaan helpottaa. Toimeksiantaja rakensi tästä yksinkertaisen muovisen kotelon, johon on asennettu neljä RJ45-naarasliitintä. RJ45-naarasliittimet kytketään kuorittuun IDE-kaapelin päähän. IDE-kaapelin toiseen päähän jätetään liitin, jolla liittinkotelo voidaan liittää Raspberry Pi:hin. Kotelo ja siihen liitetty Raspberry Pi on esitetty kuviossa 28. Kuviossa näkyy myös kaksi AM2302-anturia.



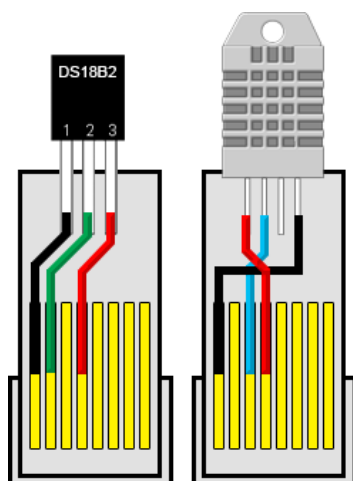
Kuvio 28 Järjestelmän kotelo ja anturit

Raspberry Pi:n GPIO 22 otettiin DS18B20-antureita varten. AM2302-antureita varten otettiin neljä GPIO-pinniä: 4, 14, 15 ja 17. GPIO:t liitetään rj45-liittimen pinneihin kuvion 29 periaatteen mukaisesti. 1-Wireä varten varattu GPIO-pinni liitetään jokaisen rj45-liittimen pinniin 6, sen lisäksi jokainen AM2302-anturia varten varattu GPIO-pinni saa oman rj45-liittimen. AM2302-antureita varten varatut GPIO:t liitetään rj45-liittimien pinneihin 7. 3,3V jännite kytketään rj45-liittimien pinneihin 4 sekä GND liitetään pinneihin 8. Näin jokaiseen rj45-liittimeen voidaan kytkeä kumpi tahansa anturi.



Kuvio 29 Järjestelmän liittimien suunnitelma

Anturit rakennetaan myös omiin rj45-naarasliittimiin, jolloin antureita voidaan liittää järjestelmään käyttäen standardi CAT-kaapeleita anturin ja liittimien välillä. Liitimestä työssä käytetään pinnejä 4, 5, 7 ja 8, joka varmistaa, että niin suoraan kun ristiin kytkettyä CAT-kaapelia voidaan käyttää antureiden ja kotelon liittimien välissä, kts. Kuvio 30.



Kuvio 30 Antureiden liittimet

5.2. Noudettavat kohteet

5.2.1. Ilmasto

Jotta Raspberry Pi osaa kommunikoida 1-Wire antureiden kanssa aktivoidaan kernelistä moduuli, jolla saadaan Raspberry Pi käyttämään 1-Wire protokollaa valitussa GPIO-pinnissä. Aktivoiminen tapahtuu yksinkertaisesti kirjoittamalla `"dtoverlay=w1-gpio,gpiopin=$GPIO"` tyhjälle riville tiedostoon `/boot/config.txt`. Uudelleen käynnistytksen jälkeen 1-Wire antureiden tiedot löytyvät kansioista `/sys/bus/w1/devices`. Kansiossa on jokaiselle GPIO-pinniin liitetyn 1-Wire verkon anturille luotu kansio, jonka nimi on anturin ROM-tunniste.

Anturin sen hetkinen arvo saadaan lukemalla tämän kansion takaa löytyvä `w1_slave` tiedosto. Kuviossa 31 on esimerkki DS18B20-anturin viestistä, jossa näkyvissä onnistunut CRC tarkistus sekä lämpötila kirjaimen `"t"` perässä. DS18B20 esittää lämpötilan tuhannen kerrannaisena, eli oikea lämpötila arvo on [luettu luku / 1000].

```
pi@kotiraspijeah ~ $ cat /sys/bus/w1/devices/10-0008019edf55/w1_slave
34 00 4b 46 ff ff 10 10 59 : crc=59 YES
34 00 4b 46 ff ff 10 10 59 t=25750
```

Kuvio 31 DS18B20 anturin viesti

AM2302-anturit vaativat toimiakseen Adafruitin DHT -kirjaston, joka sisältää vaadittavat protokollat, joilla Raspberry Pi kykenee kommunikoidaan AM2302-antureiden kanssa. Kirjasto käännetään ja asennetaan lähdekoodista. Ensin asennetaan kääntämisen vaatimat ohjelmat.

```
sudo apt-get install build-essential python-dev python-openssl
```

Tämän jälkeen ladataan Adafruitin DHT -kirjaston lähdekoodi, käännetään ja asennetaan se.

```
sudo git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install
```

5.2.2. UPS-järjestelmän tila

APC:n valmistamien UPS:ien toimintaa ja tilaa voidaan seurata sekä ohjata liittämällä UPS:in ohjausportti tietokoneeseen. Vanhemmissa UPS-laitteissa käytetään toimintaan RS-232 -porttia ja uudemmissa USB-porttia. Samaan tapaan, riippuen järjestelmän iästä, yhteys järjestelmään otetaan SNMP-, PCNET- tai Modbus protokollalla. (Kropelin & Sibbald 2015.)

Apcupsd on vapaan lähdekoodin ohjelmisto, jolla APC:n valmistamia UPS-laitteita voidaan ohjata sekä niiden tilaa seurata käyttäen tietokonetta. Ohjelmisto tukee useimpia APC:n tuottamia UPS-laitteita ja niiden käyttämiä yhteystapoja. Apcupsd on saatavilla useille eri käyttöjärjestelmille, mukaan lukien Linux, Mac OS/X ja Microsoft Windows. Ohjelmistoa voidaan käyttää joko graafisella käyttöliittymällä tai tekstikomennoilla komentoriviltä. Työhön luotu *apc-ilmasto* -skripti kerää kuviossa 32 näkyvän tapaisesta Apcupsd:n tulosteesta eri arvoja. (Kropelin & Sibbald 2015.)

```

APC      : 001,048,1088
DATE     : Fri Dec 03 16:49:24 EST 1999
HOSTNAME : daughter
RELEASE  : 3.7.2
CABLE    : APC Cable 940-0024C
MODEL    : APC Smart-UPS 600
UPSMODE  : Stand Alone
UPSNAME  : SU600
LINEV    : 122.1 Volts
MAXLINEV : 123.3 Volts
MINLINEV : 122.1 Volts
LINEFREQ : 60.0 Hz
OUTPUTV  : 122.1 Volts
LOADPCT  : 32.7 Percent Load Capacity
BATTV    : 26.6 Volts
BCHARGE  : 095.0 Percent
MBATTCHG : 15 Percent
TIMELEFT : 19.0 Minutes

```

Kuvio 32 Apcupsd status tuloste (Kropelin & Sibbald 2015)

Apcupsd löytyy Raspbianin pakettivarastosta ja asennetaan komennolla:

```
sudo apt-get install apcupsd
```

Kirjoitetaan konfiguraatio tiedostoon perusasetukset, jolla useat APC:n UPS-laitteet saadaan ohjattaviksi.

```
nano /etc/apcupsd/apcupsd.conf
```

Asetukset:

```

#/etc/apcupsd/apcupsd.conf.
UPSNAME myups
UPSCABLE usb
UPSTYPE usb

```

Asetetaan tiedostoon */etc/default/apcupsd* seuraava asetus: *"ISCONFIGURED=yes"*.

Asetus kertoo ohjelmalle, että olemme muokanneet konfigurointi tiedostoa.

```
nano /etc/default/apcupsd
```

Käynnistetään uudelleen *apcupcd*, jotta asetukset tulevat voimaan.

```
service apcupsd restart
```


5.2.3. Muistikortin I/O

Sysstat on Sebastien Gogardin kehittämä komentoriviltä ajettavista järjestelmän suorituskyvyn ja toiminnan monitorointiin tarkoitetuista työkaluista rakentuva ohjelmistopaketti. Paketti sisältää seuraavat ohjelmistot:

- *iostat*: Kertoo suorittimen tilastotietoja, sekä laitteiden, osioiden ja verkko tiedostojärjestelmien kirjoitus/luku tapahtumien tilastotietoja.
- *mpstat*: Kertoo yksittäisten tai yhdistettyjen suorittimien tilastotietoja.
- *pidstat*: Kertoo Linux prosessien tilastotietoja.
- *sar*: Kerää, raportoi ja tallentaa järjestelmän toiminnasta tietoja (CPU, muisti, verkkolaitteet, etc)
- *sadc*: Järjestelmän toimintatietojen kerääjä, käytetään yhdessä sar:n kanssa.
- *sa1*: Kerää ja tallentaa binääridataa järjestelmän päivittäisistä datatiedostoista. Ohjaa sadc:tä.
- *sa2*: Kirjoittaa tiivistetyn raportin päivittäisistä tapahtumista ja ohjaa sar:ia.
- *sadf*: Esittää sar:lla kerättyä dataa, tukee eri esitys formaatteja (CSV, XML, etc)
- *tapestat*: Kertoo tilastotietoja nauha-asemien toiminnasta.
- *sysstat*: Ohjesivu (manpage) systatin konfigurointitiedoston asetuksille.
- *cifsiostat*: Kertoo CIFS (Common Internet Filesystem) tilastotietoja. (Oetiker 2015.)

Työssä käytettiin iostat-työkalua muistikortin I/O tietojen keräämiseksi. Iostat kertoo käyttäjälle valitulta aikaväliltä muistikortin I/O tapahtumien määrän, sekä kirjoitus ja luku nopeuksien keskiarvot. Työtä varten luotu *iostat-ilmasto* -skripti kerää iostatilta minuutin välein kuluneen minuutin aikana tapahtuneet I/O tapahtumat. *Sysstat*-ohjelmistopaketti löytyy Raspbianin pakettivarastosta ja asennetaan komennolla:

```
sudo apt-get install sysstat
```

5.3. Rpi-ilmaston asentaminen

Asennetaan loput rpi-ilmaston käyttämät ohjelmat: *rrdtool*, sekä *python*. Nämä ohjelmistot ovat ladattavissa Raspbianin pakettivarastosta:

```
sudo apt-get install rrdtool python
```

Rpi-ilmasto on paketoitu *rpi-ilmasto.zip* tiedostoksi. Siirrytään */tmp*-kansioon ja ladataan tai siirretään rpi-ilmaston sisältävä paketti sinne.

```
cd /tmp
wget rpi-ilmasto+cacti.zip TAI mv /home/pi/rpi-ilmasto+cacti.zip rpi-ilmasto+cacti.zip
```

Puretaan paketti */tmp*-kansion sisään, siirretään lopullinen *rpi-ilmaston* kansio kohteeseen */usr/local/*.

```
unzip rpi-ilmasto+cacti.zip
cd rpi-ilmasto+cacti
mv -r rpi-ilmasto /usr/local/
```

Tehdään symbolinen linkki, jotta rpi-ilmasto voidaan käynnistää ja sammuttaa mistä tahansa. Kopioidaan manuaalisivu myös niille varattuun kansioon, jotta se on luettavissa *man*-komenolla.

```
ln -s /usr/local/rpi-ilmasto/bin/rpi-ilmasto /usr/bin/rpi-ilmasto
cp /usr/local/rpi-ilmasto/doc/rpi-ilmasto.1.gz /usr/share/man/man1/
```

Varmistetaan että rpi-ilmaston skriptit ovat käyttäjän ajettavissa.

```
chmod +x /usr/local/rpi-ilmasto/bin/rpi-ilmasto
chmod +x /usr/local/rpi-ilmasto/bin/simple
chmod +x /usr/local/rpi-ilmasto/bin/null-ilmasto
chmod +x /usr/local/rpi-ilmasto/bin/ds18-ilmasto
chmod +x /usr/local/rpi-ilmasto/bin/dht-ilmasto
chmod +x /usr/local/rpi-ilmasto/bin/apc-ilmasto
chmod +x /usr/local/rpi-ilmasto/bin/iostat-ilmasto
```

Seuraavaksi valitaan rpi-ilmaston OID alipuun juuri. OID valitaan kirjoittamalla tyhjälle riville *rpi.cfg* tiedostoon ilman lainausmerkkejä "*oid \$ROOT*", missä *\$ROOT* on valittu OID. OID täytyy kirjoittaa alkavan pisteellä *[.]* ja päättyvän numeroon.

```
nano /usr/local/rpi-ilmasto/bin/rpi.cfg
```

Esimerkiksi, valitaan OID .1.10.10. Tällöin AM2302-antureita voidaan kysellä OID:lla .1.10.10.1.X, DS18B20 antureita voidaan kysellä OID:lla .1.10.10.2.X sekä UPS:n tietoja OID:lla .1.10.10.3.X.

```
# Esimerkki
oid .1.10.10
```

Konfiguroidaan **crontab**, jotta virhemittari saadaan nollaantumaan päivittäin sekä antureiden rikkoutuessa SNMP antamaan antureiden arvoksi tuntematon. Estää SNMP:tä antamasta väärää arvoja, jos anturit lakkaavat vastaamasta.

```
sudo crontab -e
```

Asetetaan seuraavat kaksi riviä crontabin pohjalle tyhjille riveille:

```
0 0 * * * /bin/bash /usr/local/rpi-ilmasto/bin/null-ilmasto 00
59 * * * * /bin/bash /usr/local/rpi-ilmasto/bin/null-ilmasto rest
```

Jotta rpi-ilmasto saadaan käynnistymään järjestelmän käynnistyessä, täytyy tyhjälle riville lisätä myös:

```
@reboot /usr/bin/rpi-ilmasto start
```

Konfiguroidaan **logrotate**, jotta rpi-ilmaston lokitiedostot eivät vahingossakaan kasva loputtoman suuriksi.

```
nano /etc/logrotate.conf
```

Asetetaan seuraavat tiedot tiedoston pohjalle tyhjille riveille.

```
/usr/local/rpi-ilmasto/log/*.log {
    monthly
    missingok
    nocreate
    minsize 1M
    maxsize 5M
    rotate 1
    notifempty
}
```

MIB-malli tiedosto löytyy rpi-ilmaston asennus paketissa olevasta kansioista nimeltä **MIB**. MIB tiedoston nimi on **RPIILMASTO-MIB.template**. Muokataan tiedoston rivillä 25 kohtaa **::= {xy-oid-xy}**. Asetetaan valitun alipuun OID siihen niin että pisteiden tilalla on välilyönnit. Esim .1.10.10 -> ::= { 1 10 10 }.

```
cd /tmp/rpi-ilmasto+cacti/MIB
nano RPIILMASTO-MIB.template
```

Nimetään MIB-malli *"RPIILMASTO-MIB.txt"*:ksi ja siirretään se käytetyn SNMP managerin MIB kansioon. NET-SNMP:n oletuskansio MIB-moduuleille Rasbian-käyttöjärjestelmässä on *"/usr/share/snmp/mibs/"*.

```
mv RPIILMASTO-MIB.template /usr/share/snmp/mibs/RPIILMASTO-MIB.txt
```

MIB-moduuli täytyy myös aktivoida. NET-SNMP:n managerille aktivointi tapahtuu muokkaamalla tiedostoon *"/etc/snmp/snmp.conf"* rivi *"mib +RPIILMASTO-MIB"*. Poistetaan tai kommentoidaan myös rivi *"mibs :"*, joka ottaisi MIBien lataamisen pois käytöstä.

```
nano /etc/snmp/snmp.conf
```

Kun moduuli on asennettu managerille, voidaan esimerkiksi DS18B20 antureiden virhelaskuri noutaa nimellä *"ds18-errors"* (kts. Kuvio 33), sen sijaan että käytetään OID:ta *"\$ROOT.2.0"*. Kaikki eri nimet ovat luettavissa liitteessä 2. Noudettavat nimet määritetään kehyksillä MODLUE-IDENTITY, OBJECT-IDENTIFIER ja OBJECT-TYPE.

```
RPIILMASTO-MIB::ds18-errors = STRING: "0"
~ $ snmpget -v2c -c rpiuser localhost ds18-errors
```

Kuvio 33 DS18 virheiden SNMP haku sanalla

5.4. SNMP-agentti

Net-SNMP on avoimenlähdekoodin SNMP-ohjelmistopaketti. Paketti tukee kaikkia SNMP:n versioita. Useat pakettiin kuuluvat ohjelmat ajetaan komentoriviltä. Ohjelmistopaketti sisältää muun muassa: (Shield 2011.)

- Komennot kaikille SNMP hallinta-aseman toiminnoille (Get, Getnext, Set) sekä edistyneempiä joukkokyselyjä (Walk).
- Ilmoituksia (Trap) vastaanottavan ohjelman. Ilmoituksia voidaan kirjata lokeihin tai lähettää edelleen muille ohjelmille.
- Laajennettavan SNMP hallinta-agentti ohjelman. (Shield 2011.)

Net-SNMP ohjelmistopakettien hallinta-agentti on nimeltään snmpd. Hallinta-agentti ajetaan taustaohjelmana, missä se vastaanottaa ja vastaa SNMP hallintaviesteihin.

Snmpd:n toimintaa voidaan laajentaa muutamilla eri tavoilla, työssä laajentaminen tehtiin *“pass”* komennolla. Komento luovuttaa OID alipuun hallinnan siihen asetetulle ohjelmalle. Luovuttaminen tapahtuu lähettämällä tähän OID:hin osoitetut kyselyt asetetulle ohjelmalle ja luomalla ohjelman antamista tulosteista vastaus.

Kyselyt lähetetään ohjelmalle 2–4 muuttujan kanssa: (SNMPD.CONF 2002.)

1. SNMP kysely/komento.
 - -g = GetRequest.
 - -n = GetNextRequest.
 - -s = SetRequest.
2. Kohteen yksilöintitunnus (OID).
3. Jos komento on -s, asetettavan arvon/viestin tyyppi.
4. Jos komento on -s, asetettava arvo/viesti.

Komennoilla *“-g”* ja *“-n”* ohjelman täytyy vastata snmpd:lle kolmella viestillä.

1. Vastattavan kohteen yksilöintitunnus (OID).
2. Vastauksen arvon/viestin tyyppi.
3. Vastaus arvo/viesti.

Jos kysely (-g, -n) yksilöintitunnus ei ole olemassa tai on virheellinen, ohjelman täytyy sulkeutua ilman viestiä. Jos snmpd ei saa viestiä, kyselyyn vastataan *“NoSuchName”*.

Komennolla *“-s”*, ohjelman täytyy onnistuessaan sulkeutua ilman viestiä. Jos snmpd ei saa viestiä, kirjoituksen oletetaan onnistuneen. Jos kirjoitus on virheellinen, ohjelman täytyy ilmoittaa siitä snmpd:lle viestillä *“not-writable”* tai *“wrong-type”*. (SNMPD.CONF 2002.)

Snmpd löytyy Raspbianin pakettivarastosta ja asennetaan komennolla:

```
apt-get install snmpd
```

Konfiguroidaan *snmpd* kirjoittamalla ilman lainausmerkkejä rivi "*pass \$ROOT /bin/bash /usr/local/rpi-ilmasto/bin/simple*" tiedostoon */etc/snmp/snmpd.conf*. *\$ROOT* on sama OID, joka asetettiin *rpi-ilmaston* konfigurointitiedostoon *rpi.cfg*.

```
nano /etc/snmp/snmpd.conf
```

Kirjoitetaan myös tarkempia asetuksia. Seuraavilla asetuksilla *snmpd* kuuntelee porttia 161 (oletusportti) sekä luo SNMPv2 käyttäjän "*rpiuser*", jolla on lukuoikeudet valittuun OID:hin sekä muutamiin laitetietoihin: Prosessorin kuorma (~2021.10.1.3), muistintiedot (~2021.4), levyntila (~2021.9.1) sekä toiminta-aika (~2.1.1.3).

```
#####
# Rpi-configuration      #
#####
agentAddress 161
com2sec rpisecurity default rpiuser

view rpireadonly included .1.3.6.1.4.1.2021.10.1.3
view rpireadonly included .1.3.6.1.4.1.2021.4
view rpireadonly included .1.3.6.1.2.1.1.3
view rpireadonly included .1.3.6.1.4.1.2021.9.1
view rpireadonly included .1.3.6.1.2.1.1.1.0
view rpireadonly included $ROOT

group rpigroup v2c rpisecurity
access rpigroup "" any noauth prefix rpireadonly none none

pass $ROOT /bin/bash /usr/local/rpi-ilmasto/bin/simple
disk / 100000
sysServices 72
```

Muokkausten jälkeen *snmpd* käynnistetään uudelleen.

```
service snmpd restart
```

Snmpd:n ja *rpi-ilmaston simple*-skriptin toiminta tarkistetaan suorittamalla *GetRequest* valitulla *\$ROOT* OID:lla.

```
apt-get install snmp
snmpget -v2c -c rpiuser localhost $ROOT
```

Jos konfigurointi on tehty oikein ja toimii, saadaan vastaukseksi:

```
STRING: rpi-ilmasto-system
```

6. Valvonta

6.1. Mokkulan käyttöönotto

Raspberry Pi:lle saadaan tekstiviesti ominaisuudet Huawei Technologiesin valmistamalla E220-mallisella 3G-modeemilla. E220-malliset modeemit ovat oletuksena muistitilassa, jolla valmistaja varmistaa, että laitteen ajurit ovat aina saatavilla käyttäjälle. Moodi saadaan vaihdettua asentamalla *usb-modeswitch* -ohjelmisto, asennuksen jälkeen moka vaihtaa järjestelmän käynnistyksen yhteydessä moodin mobiililaitteeksi.

```
sudo apt-get install usb-modeswitch
```

Mokkulalla lähetetään tekstiviestit käyttäen *gammu-smsd* -ohjelmistoa. *Gammu-smsd* on Linuxin taustalla pyörivä ohjelma, joka lähettää tekstiviestejä. Lähetettävät viestit noudetaan *output*-kansioista ja lähetetään edelleen tekstiviestinä. Käyttäjä voi lisätä lähetettäviä viestejä *output*-kansioon *gammu-smsd-inject* -sovelluksella.

```
sudo apt-get install gammu-smsd
```

Ohjelman asennuksen jälkeen muokataan konfiguraatiotiedostoa */etc/gammu-smsdrc* ja asetetaan sinne seuraavat tiedot:

```
[gammu]
# Please configure this!
port = /dev/ttyUSB0
connection = at19200
```

Koe tekstiviestin voi lähettää komennolla:

```
echo $viesti | gammu-smsd-inject TEXT $puhelinnumero
```

6.2. Cactin Asentaminen

Raspberry Pi:lle asennettiin Cactin uusin versio, joka asennus hetkellä oli 0.8.8f. Cacti vaatii että järjestelmään on asennettu seuraavat ohjelmistot:

- RRDTool, 1.0.49 tai uudempi.
- MySQL 5.0 tai uudempi.
- PHP 5.1 tai uudempi.
- WWW-palvelu, PHP tuella.

Cactin asentaminen aloitetaan asentamalla *apache2* WWW-palvelin ohjelmisto.

```
apt-get install apache2
```

Seuraavaksi asennetaan Cactin vaatimat riippuvuudet ja apachen vaatimat laajennukset: *mysql-server*, *snmp*, *php5*, *php5-snmp*, *php5-mysql* ja *rrdtool*.

```
apt-get install mysql-server snmp php5 php5-snmp php5-mysql rrdtool
```

Jos MySQL:n asennuksen aikana ei aseteta *root* käyttäjän salasanaa, ajetaan seuraava komento.

```
mysql_secure_installation
```

Siirrytään kansioon */tmp*, ladataan Cactin uusin tarball, puretaan se ja siirrytään Cactin tiedostot sisältävään kansioon.

```
cd /tmp
wget www.cacti.net/downloads/cacti-0.8.8f.tar.gz
tar -zxvf cacti-0.8.8f.tar.gz
cd cacti-0.8.8f
```

Luodaan MySQL:ään tietokanta *cacti* ja tuodaan Cactin mukana tullut tietokanta sinne.

```
mysqladmin --user=root -p create cacti
mysql --user=root -p cacti < cacti.sql
```

Luodaan Cactia varten MySQL käyttäjä sekä annetaan tälle käyttäjälle oikeudet juuri luotuun *cacti*-tietokantaan. Tässä käyttäjä nimeksi annetaan *cactiusername* sekä salasanaksi asetetaan *cactipassword*.

```
mysql --user=root -p
> CREATE USER 'cactiusername'@'localhost' IDENTIFIED BY 'cactipassword';
> GRANT ALL PRIVILEGES ON cacti. * TO 'cactiusername'@'localhost';
> quit
```


Siirrytään yksi askel taaksepäin. Siirretään Cactin kansio WWW-palvelimen juureen ja poistetaan nimestä kaikki sanan *cacti* jälkeen.

```
cd ..
sudo mv cacti-0.8.8f /var/www/html/cacti
```

Siirrytään Cactin pääkansioon, muokataan asetuksia kohteesta *cacti/include/config.php*. Osoitetaan tiedostoon juuri luodut MySQL käyttäjä ja tietokanta tiedot.

```
cd /var/www/html/cacti
nano include/config.php
```

Tiedoston pitäisi näyttää seuraavan näköiseltä muokkauksen jälkeen.

```
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cactiuser"
$database_password = "cactipassword"
```

Annetaan apache2:n käyttäjälle oikeudet Cactin kansioihin ja tiedostoihin.

```
sudo chown -R www-data:www-data /var/www/html/cacti
find /var/www/cacti -type f -exec chmod 664 {} \;
find /var/www/cacti -type d -exec chmod 775 {} \;
```

Cacti noutaa tiedot sille osoitetuista kohteista ajamalla *poller.php* -skriptin *crontabilla*. Oletuksena skripti ajetaan 5 minuutin välein, järjestelmä asennettiin hakemaan arvot minuutin välein. Avataan järjestelmän laajuinen *crontab* konfiguraatiotiedosto ja lisätään sinne rivi:

```
* * * * * www-data php /var/www/html/cacti/poller.php > /dev/null 2>&1
```

```
nano /etc/crontab
```

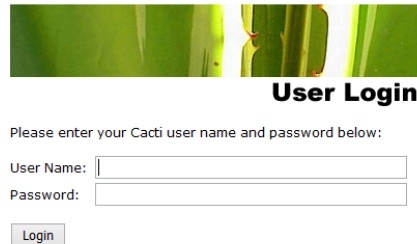
Ladataan Cactille kaksi laajennusta, *thold* sekä sen vaatima *settings*. Puretaan ne Cactin pääkansioista löytyvään *plugins*-kansioon.

```
cd /var/www/cacti/plugins
docs.cacti.net/_media/plugin:settings-v0.71-1.tgz
wget docs.cacti.net/_media/plugin:thold-v0.5.0.tgz
tar -zxvf settings-v0.71-1.tgz
tar -zxvf thold-v0.5.0.tgz
```

6.3. Cactin asetukset

Avataan Cactin käyttöliittymä käyttäen WWW-selainta: <http://sinun-osoite/cacti/>.

Oletuksena pääkäyttäjä on **"admin"** ja sen salasana on **"admin"**, kirjautuessasi sisään joudutaan vaihtamaan salasana. Kuviossa 34 sisäänkirjautumisikkuna.



User Login

Please enter your Cacti user name and password below:

User Name:

Password:

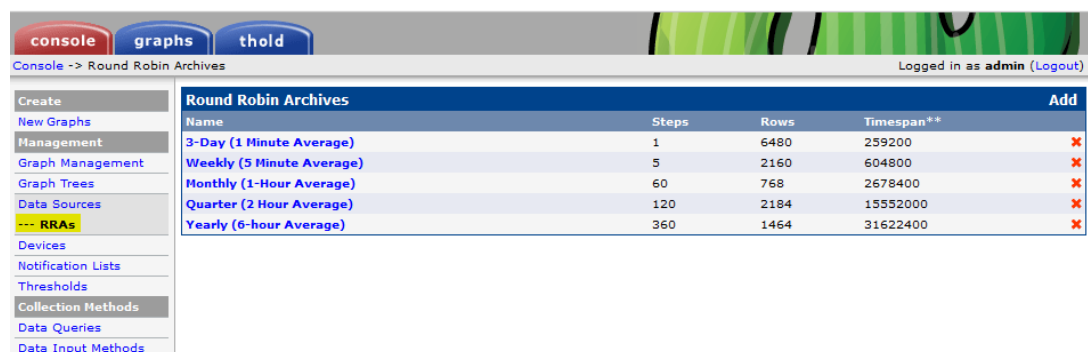
Kuvio 34 Cactin kirjautumis ikkuna

Jotta Cacti saamaan tukemaan 1-minuutin päivitysväliä täytyy ensimmäiseksi

muokata Cactin luomien RRD tietokantojen asetuksia. Asetukset löytyvät

"Management" tunnisteiden alta, valitsemalla **"Data Sources"** ja sen alta **"--- RRAs"**.

Kuviossa 35 on asetusten sijainti, sekä uudet arkistojen asetukset, kuvioon on myös maalattu mistä kuvion näkymän löytää.



Name	Steps	Rows	Timespan**	Add
3-Day (1 Minute Average)	1	6480	259200	✗
Weekly (5 Minute Average)	5	2160	604800	✗
Monthly (1-Hour Average)	60	768	2678400	✗
Quarter (2 Hour Average)	120	2184	15552000	✗
Yearly (6-hour Average)	360	1464	31622400	✗

Kuvio 35 Cactin rrd tietokantojen arkistojen astukset

Arkistojen nimiin on merkitty millä ajantarkkuudella arvoja tallennetaan. Tietokantaan lisätään arvoja minuutin välein. Arkistoihin tallennetaan arvoja ajanjaksolta, joka on $[1\text{min} \cdot \text{steps} \cdot \text{rows}]$. Taulukkoon 4 on laskettu kuinka pitkältä ajalta arkistot ylläpitävät arvoja. Kuviossa 35 näkyvä *Timespan* ei osoita arkistojen kokoa, vaan valitsee miltä ajanjaksolta sekunneissa tämän arkiston kuviot oletuksena piirretään.

Taulukko 4 Cactin rrd tietokantojen arkistojen asetusten ajanjaksot

	Step	Rows	Kirjattava Ajanjakso	
3-Day (1 Minute Average)	1	6480	4,5	päivää
Weekly (5 Minute Average)	5	2160	7,5	päivää
Monthly (1 Hour Average)	60	768	32	päivää
Quarter (2 Hour Average)	120	2184	182	päivää
Yearly (6 Hour average)	360	1464	366	päivää

Seuraavaksi siirrytään asetuksiin ja asetetaan *cmd.php* pollerin päivitysjakso yhteen minuuttiin. Asetukset löytyvät ”*Configuration*” tunnisteiden alla olevan ”*Settings*” osion takaa. *Settings*issä siirrytään *Poller* välilehteen ja vaihdetaan *Poller Interval* sekä *Cron Interval* asetukseen *Every Minute*. Kuviossa 36 näkyy asetukset ja siihen on maalattu mainitut kohdat.

The screenshot shows the Cacti web interface for configuring the poller. The left sidebar contains a navigation menu with options like 'Create', 'New Graphs', 'Management', 'Graph Management', 'Graph Trees', 'Data Sources', 'Devices', 'Notification Lists', 'Thresholds', 'Collection Methods', 'Data Queries', 'Data Input Methods', 'Templates', 'Graph Templates', 'Host Templates', 'Data Templates', 'Threshold Templates', 'Import/Export', 'Import Templates', 'Export Templates', 'Configuration', 'Settings', 'Plugin Management', 'Utilities', 'System Utilities', 'User Management', and 'Logout User'. The 'Settings' option is highlighted. The main content area is titled 'Cacti Settings (Poller)' and has several tabs: 'General', 'Paths', 'Poller', 'Graph Export', 'Visual', 'Authentication', 'Mail / DNS', and 'Thresholds'. The 'Poller' tab is active. Under the 'General' section, there are several settings: 'Enabled' (checked), 'Poller Type' (set to 'cmd.php'), 'Poller Interval' (set to 'Every Minute'), 'Cron Interval' (set to 'Every Minute'), 'Maximum Concurrent Poller Processes' (set to 1), 'Balance Process Load' (checked), 'Spine Specific Execution Parameters' (a sub-section), 'Maximum Threads per Process' (set to 1), 'Number of PHP Script Servers' (set to 1), and 'Script and Script Server Timeout Value' (set to 10).

Kuvio 36 Poller asetusten asettaminen cactiin

Viimeisenä täytyy kaikkien datamallien asetuksia muokata, jotta niitä voidaan seurata yhden minuutin tarkkuudella. Datamallit löytyvät ”*Templates*” tunnisteiden alla olevan ”*Data Templates*” osion takaa. Jokaisessa datamallissa muokataan asetukset *Associated RRA’s*, *Step* sekä *Heartbeat*. *Associated RRA’s* asetukseen maalataan kaikki arkistot, joita halutaan käyttää. Järjestelmässä suurin osa kohteista lisätään käyttämällä *SNMP - Generic OID Templatea*, joten sille asetetaan kaikki arkistot käyttöön. *Step* kertoo kuinka usein malli päivitetään, se asetetaan yhteen minuuttiin. *Heartbeat* kertoo kuinka kauan odotetaan, ennen kuin arvo asetetaan tuntemattomaksi. *Heartbeatin* asetukseksi asetetaan kaksi kertaa *Stepin* arvo. Kuviossa 37 on avattu *SNMP - Generic OID Template* ja maalattu mainitut asiat. *SNMP - OID Template*ssa valitaan myös kuviossa 37 punaisella merkityt asetukset, jotta niitä voidaan muuttaa, kun SNMP kohteita lisätään seurattaviksi.

The screenshot shows the Nagios Core web interface for editing the 'SNMP - Generic OID Template'. The left sidebar contains a navigation menu with options like 'Create', 'New Graphs', 'Management', 'Graph Management', 'Graph Trees', 'Data Sources', 'Devices', 'Notification Lists', 'Thresholds', 'Collection Methods', 'Data Queries', 'Data Input Methods', 'Templates', 'Graph Templates', 'Host Templates', 'Data Templates' (highlighted), 'Threshold Templates', 'Import/Export', 'Import Templates', 'Export Templates', 'Configuration', 'Settings', 'Plugin Management', 'Utilities', 'System Utilities', 'User Management', and 'Logout User'.

The main content area is titled 'Data Templates [edit: SNMP - Generic OID Template]'. It contains the following sections:

- Name:** The name given to this data template. Value: 'SNMP - Generic OID Template'.
- Data Source:**
 - Name:** Value: '|host_description| -'.
 - Use Per-Data Source Value (Ignore this Value):** ☒
 - Data Input Method:** Value: 'Get SNMP Data'.
 - Associated RRA's:** Value: '3-Day (1 Minute Average)', 'Weekly (5 Minute Average)', 'Monthly (1-Hour Average)', 'Quarter (2 Hour Average)'.
 - Step:** Value: '60'.
 - Use Per-Data Source Value (Ignore this Value):** ☒
 - Data Source Active:** ☒ Data Source Active.
- Data Source Item [snmp_oid]:** (New)
 - Internal Data Source Name:** Value: 'snmp_oid'.
 - Use Per-Data Source Value (Ignore this Value):** ☐
 - Minimum Value ('U' for No Minimum):** Value: '0'.
 - Use Per-Data Source Value (Ignore this Value):** ☒
 - Maximum Value ('U' for No Maximum):** Value: 'U'.
 - Use Per-Data Source Value (Ignore this Value):** ☒
 - Data Source Type:** Value: 'GAUGE'.
 - Use Per-Data Source Value (Ignore this Value):** ☒
 - Heartbeat:** Value: '120'.
 - Use Per-Data Source Value (Ignore this Value):** ☒

Kuvio 37 Cactin Data templateiden asetukset

Työtä varten jouduttiin luomaan muutamia uusia kuviomalleja (*Graph Template*). Kuviossa 38 on näkyvissä niistä yksi. Kaikilla muilla luoduilla malleilla on samat asetukset, mutta eri määrin. Malleja tehdään niin, että on mahdollista tehdä kuviot jotka sisältävät yhdestä kuuteen eri datalähdettä.

Create	Graph Template Items [edit: 2 Sources - Generic Full]						Add
New Graphs	Graph Item	Data Source	Graph Item Type	CF Type	Item Color		
Management	Item # 1	(snmp_oid):	LINE2	LAST	FF0000	↕ ↗	✖
Graph Management	Item # 2	(snmp_oid):	GPRINT	LAST		↕ ↗	✖
Graph Trees	Item # 3	COMMENT: <HR>	COMMENT	AVERAGE		↕ ↗	✖
Data Sources	Item # 4	(snmp_oid):	LINE2	LAST	00CF00	↕ ↗	✖
Devices	Item # 5	(snmp_oid):	GPRINT	LAST		↕ ↗	✖
Notification Lists	Item # 6	COMMENT: <HR>	COMMENT	AVERAGE		↕ ↗	✖
Thresholds	Item # 7	COMMENT:	COMMENT	AVERAGE		↕ ↗	✖
Collection Methods	Graph Item Inputs						Add
Data Queries	Name						
Data Input Methods	1. CDEF						✖
Templates	1.1 Legend Color						✖
Graph Templates	1.1.1 Graph Type						✖
Host Templates	1.2 Data Source						✖
Data Templates	1.3 Legend Text						✖
Threshold Templates	1.4 After Value Comment						✖
Import/Export	2. CDEF						✖
Import Templates	2.1 Legend Color						✖
Export Templates	2.1.1 Graph Type						✖
Configuration	2.2 Data Source						✖
Settings	2.3 Legend Text						✖
Plugin Management	2.4 After Value Comment						✖
Utilities	99. Last Comment						✖
System Utilities							✖

Kuvio 38 Cactin Graph Templaten asetukset

Kuviossa 38 olevalla mallilla voidaan valita kaksi datalähdettä, joista piirretään kaksi käyrää. ”Graph Item Inputs” osiossa määritetään asiat, joita käyttäjä voi muokata käyttäessään kyseistä mallia:

- *CDEF*: Muokkaus funktio, jolla arvoja voidaan muokata ennen piirtämistä kuvioon.
- *Legend Color*: Käyrän väri.
- *Graph Type*: Käyrän tyyppi. (Käyrä, alue, teksti).
- *Legend Text*: Teksti ennen arvoa.
- *After Value Comment*: Teksti arvon jälkeen.
- *Last Comment*: Ylimääräinen rivi, johon voi kirjoittaa tekstiä.

6.4. Cactin laajentaminen

Cactille asennetaan laajennukset *settings* ja *thold* (Treshold). Thold on laajennus, jolla seurattaville kohteille voidaan asettaa raja-arvoja, joiden ylittyessä lähetetään sähköpostiin hälytys. Settings on perustoimintolaajennus, jonka thold vaatii toimiakseen. Toimeksiantaja ei halunnut sähköpostilaajennusta, joten tholdin toimintaa laajennettiin edelleen muokkauksella, joka mahdollistaa hälytyksen lähettämisen tekstiviestillä.

Laajennukset aktivoidaan *"Configuration"* tunnisteiden alla olevan *"Plugin Management"* osion takaa. *"Actions"* sarakkeen alla olevilla ikoneilla laajennuksia asennetaan ja aktivoidaan.

Tholdia on muokattu muidenkin toimesta tekemään tekstiviestihälytyksiä.

Nimimerkki HendryLeo on kirjoittanut tholdiin muokkauksen, joka mahdollistaa Cactin lähettämään käyttäjälle tekstiviesti ilmoituksen, jos se menettää yhteyden johonkin seurattavaksi asetetuista laitteista. Hänen työtään muokattiin edelleen niin, että tekstiviestin voi lähettää myös kun asetetut raja-arvot ylitetään. (Gammu integration with Cacti Threshold 2012)

Tekstiviesti hälytyksen saamiseksi tholdin kolmea tiedostoa muokattiin.

Ensimmäiseksi luotiin asetukset, joilla käyttäjä voi aktivoida tekstiviesti hälytyksen ja asettaa puhelinnumeron, johon viesti lähetetään. Seuraavat rivit lisättiin tiedostoon *thold/included/settings.php*.

```
// RPI-ILMASTO luodaan sms asetukset
'thold_sms_header' => array(
    'friendly_name' => 'SMS Options',
    'method' => 'spacer',
),
// --
'thold_gammu_smsd_inject_path' => array(
    'friendly_name' => 'Path to gammu-smsd-inject',
    'description' => 'This is the path to gammu sms inject binary',
    'method' => 'textbox',
    'max_length' => 255,
),
'thold_sms_number' => array(
    'friendly_name' => 'SMS Contact',
    'description' => 'This is the SMS Number to contact. Separated by semicolon
(;) ',
    'method' => 'textbox',
    'max_length' => 255,
),
'thold_send_sms' => array(
    'friendly_name' => 'Send Dead Hosts Notifications as SMS',
    'description' => 'If checked, this will cause Host Up/Down notices to be sent as
SMS.',
    'method' => 'checkbox',
    'default' => 'off'
),
'thold_alert_sms' => array(
    'friendly_name' => 'Send all threshold Alerts as SMS',
    'description' => 'If checked, this will cause all threshold triggered Alerts to be
sent as SMS.',
```

```
'method' => 'checkbox',
'default' => 'off'
),
// RPI
```

Kuviossa 39 näytetään kuinka asetetut rivit näkyvät käyttäjälle tholdin asetuksissa.

Kuvio 39 Tekstiviesti hälytysten asetukset Tholdissa

Seuraavaksi muokattiin *thold/includes/polling.php* tiedostoa. Tiedostoon lisättiin kolmeen kohtaa rivejä. Seuraavan rivin jälkeen:

```
$alert_email = read_config_option('alert_email');
```

Lisättiin asetusten lukeminen, rivi noutaa aikaisemmin luodun asetuksen:

```
// RPI-ILMASTO Luetaan sms asetus
$alert_sms = read_config_option('thold_send_sms');
// RPI
```

Kahteen paikkaan rivien alle:

```
thold_mail($alert_email, "", $subject, $msg, "");
}
```

Kirjoitettiin rivit, jotka lähettävät tekstiviesti hälytyksen, jos Cacti ei saa yhteyttä sinne asetettuun laitteeseen.

```
// RPI-ILMASTO
if ($alert_sms == 'on') {
    thold_sms($subject);
}
```

Viimeisenä muokattiin *thold/thold_functions.php* tiedostoa. Tiedostoon asetettiin muutamia rivejä useaan eri kohtaan. Rivin alle, jolle on kirjoitettu seuraavaa:

```
/* pull a few default settings */
```

Asetettiin seuraavat rivit, rivi noutaa aikaisemmin luodun asetuksen.

```
// RPI-ILMASTO Luetaan sms asetus
$alert_sms = read_config_option('thold_alert_sms');
// RPI
```

Seuraavien rivien alle:

```
thold_mail($alert_emails, "", $subject, $msg, $file_array);
}
```

Kirjoitettiin seuraavat rivit. Rivit laukaisevat hälytyksen, jos tekstiviesti raja-arvo hälytys on asetettu päälle. Rivit lisättiin seitsemään kohtaan.

```
// RPI-ILMASTO
if ($alert_sms == 'on') {
    thold_sms($subject);
}
// RPI
```

Seuraavaksi luotiin tekstiviestin lähetyskomento. Rivit lisättiin tiedoston viimeiseksi. Hälytysviestinä käytetään sähköpostin aiheena käytettyä tekstijonoa. Tekstiviesti lähetetään käyttäen *gammu-smsd* -ohjelmistoa.

```
// RPI-ILMASTO lisataan komento jolla voidaan lähettää sms

function thold_sms($msg) {
    global $debug;
    thold_debug('Preparing to send SMS message');
    $gammu_output = array();
    $gammu_smsd_inject_path = trim(read_config_option('thold_gammu_smsd_inject_path'));
    $sms = trim(read_config_option('thold_sms_number'));
    $sms_numbers = explode(':', $sms);
    foreach($sms_numbers as $key => $value) {
        $gammu_smsd_inject_command = $gammu_smsd_inject_path . ' EMS ' . trim($value) . ' -text "'
        . $msg . '"';
        exec($gammu_smsd_inject_command, $gammu_output, $gammu_return);

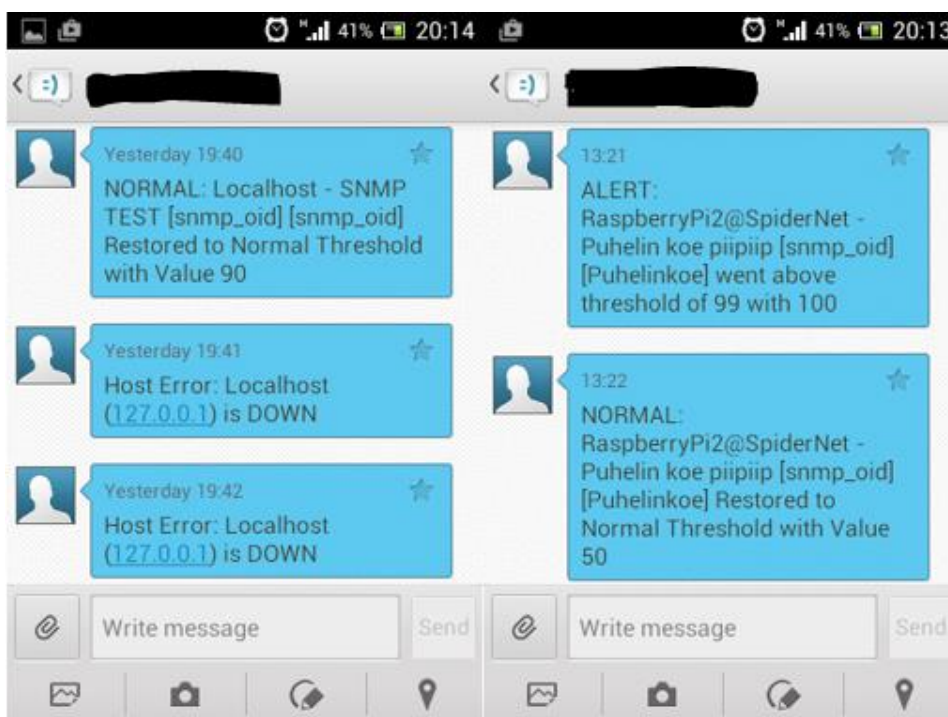
        if (read_config_option("log_verbosity") >= POLLER_VERBOSITY_DEBUG || $debug) {
            cacti_log('DEBUG: gammu-sms-inject called with command ==>' .
            $gammu_smsd_inject_command, true, 'POLLER');
            foreach ($gammu_output as $key => $value) {
                cacti_log('DEBUG: gammu-sms-inject output ==>' . $key . '=' . $value, true, 'POLLER');
            }
            cacti_log('DEBUG: gammu-sms-inject return value ==>' . $gammu_return, true, 'POLLER');
        }
    }
}

// RPI
```


Viimeisenä asetetaan apachen luomalle käyttäjälle oikeus lähettää tekstiviestejä. Oikeudet annetaan asettamalle käyttäjä *dialout*-ryhmään sekä antamalla oikeudet *gammu-smsd-inject* -ohjelman käyttämiseen.

```
sudo adduser www-data dialout
sudo usermod -a -G dialout www-data
sudo chmod 4755 /usr/bin/gammu-smsd-inject
```

Thold:ille on nyt asetettu kyky lähettää tekstiviestejä. Kuviossa 40 näkyy muutama Cactin lähettämä tekstiviesti, kun toimintoa koestettiin. Alkaen vasemmalta, alaspäin: Laitteella "Localhost" seurattava kohde "SNMP TEST" on palautunut normaaliin arvolla 90. Sen alla kaksi viestiä, jotka ilmoittavat että Cacti on menettänyt yhteyden laitteeseen "Localhost". Oikealla kaksi viestiä, joissa seurattava kohde "SpiderNet – Puhelin koe piipiip" menee yli raja-arvon ja palautuu minuutin kuluttua normaaliin tilaan.



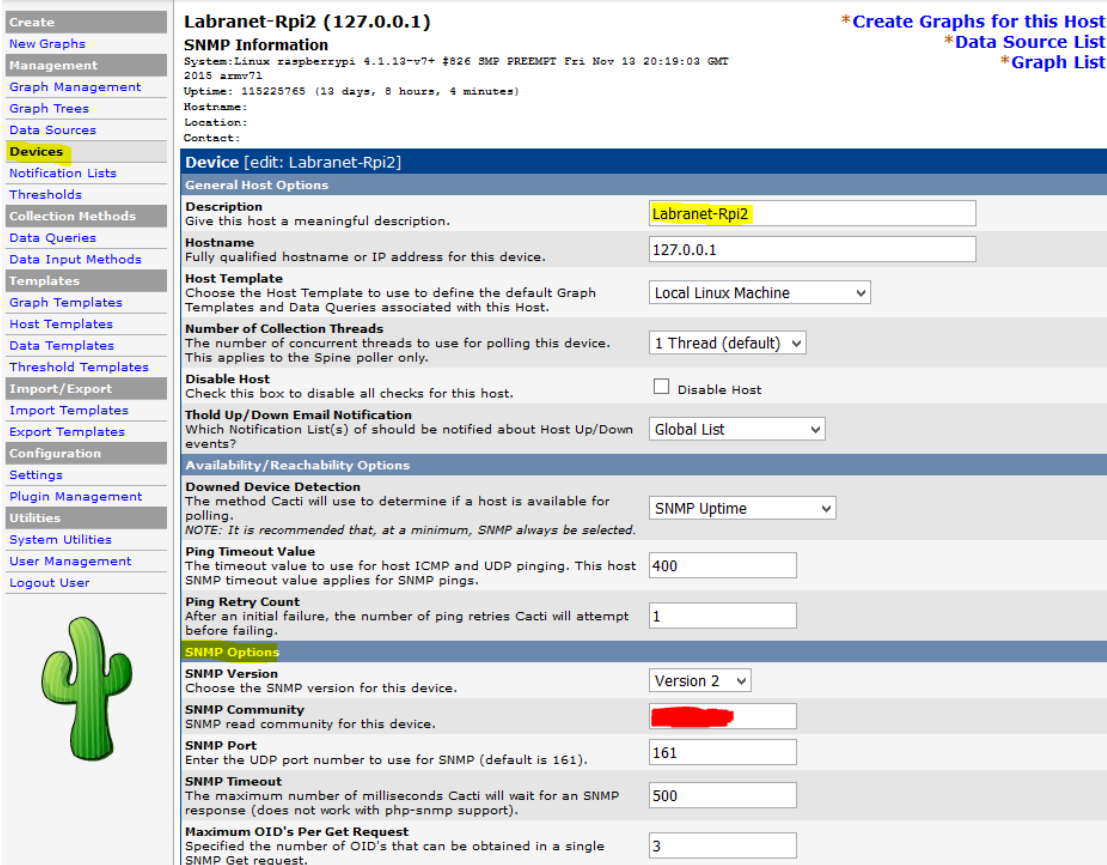
Kuvio 40 Tekstiviesti hälytyksen koeastaminen

6.5. Kohteiden lisääminen

Cactille asetetaan laite (*Device*), jolle asetetaan datalähteitä (*Data Source*), joilla kerätyistä arvoista piirretään käyttäjälle kuvioita (*Graph*).

6.5.1. Laite

Oletuksena Cactin isäntälaitte on asetettuna laitteeksi. Laitteet löytyvät ”*Management*” tunnisteesta alta, ”*Devices*”-osiosta. Kuviossa 41 on avattu *localhost* ja muutettu sille nimeksi *Labranet-Rpi2*, tunnistamisen helpottamiseksi. Laitteeseen asetetaan myös oletus SNMP asetukset, näin ollen niitä ei tarvitse joka kerta erikseen asettaa, kun seurattavia kohteita aletaan lisäämään.



The screenshot shows the Cacti web interface for configuring a device. The left sidebar contains a navigation menu with options like 'Create', 'New Graphs', 'Management', 'Graph Management', 'Graph Trees', 'Data Sources', 'Devices' (highlighted), 'Notification Lists', 'Thresholds', 'Collection Methods', 'Data Queries', 'Data Input Methods', 'Templates', 'Graph Templates', 'Host Templates', 'Data Templates', 'Threshold Templates', 'Import/Export', 'Import Templates', 'Export Templates', 'Configuration', 'Settings', 'Plugin Management', 'Utilities', 'System Utilities', 'User Management', and 'Logout User'. Below the menu is a green cactus icon.

The main content area is titled 'Labranet-Rpi2 (127.0.0.1)' and includes a 'SNMP Information' section with details like 'System: Linux raspberrypi 4.1.13-v7+ #826 SMP PREEMPT Fri Nov 13 20:19:03 GMT 2015 armv7l' and 'Uptime: 115225765 (13 days, 8 hours, 4 minutes)'. Below this is the 'Device [edit: Labranet-Rpi2]' configuration form.

The configuration form is divided into several sections:

- General Host Options:**
 - Description:** A text field containing 'Labranet-Rpi2'.
 - Hostname:** A text field containing '127.0.0.1'.
 - Host Template:** A dropdown menu set to 'Local Linux Machine'.
 - Number of Collection Threads:** A dropdown menu set to '1 Thread (default)'.
 - Disable Host:** A checkbox labeled 'Disable Host' which is unchecked.
 - Thold Up/Down Email Notification:** A dropdown menu set to 'Global List'.
- Availability/Reachability Options:**
 - Downed Device Detection:** A dropdown menu set to 'SNMP Uptime'.
 - Ping Timeout Value:** A text field containing '400'.
 - Ping Retry Count:** A text field containing '1'.
- SNMP Options:**
 - SNMP Version:** A dropdown menu set to 'Version 2'.
 - SNMP Community:** A text field containing a redacted value.
 - SNMP Port:** A text field containing '161'.
 - SNMP Timeout:** A text field containing '500'.
 - Maximum OID's Per Get Request:** A text field containing '3'.

On the right side of the configuration form, there are links: '* Create Graphs for this Host', '* Data Source List', and '* Graph List'.

Kuvio 41 Laite asetukset Cactissa

6.5.2. Seurattava asia/olio

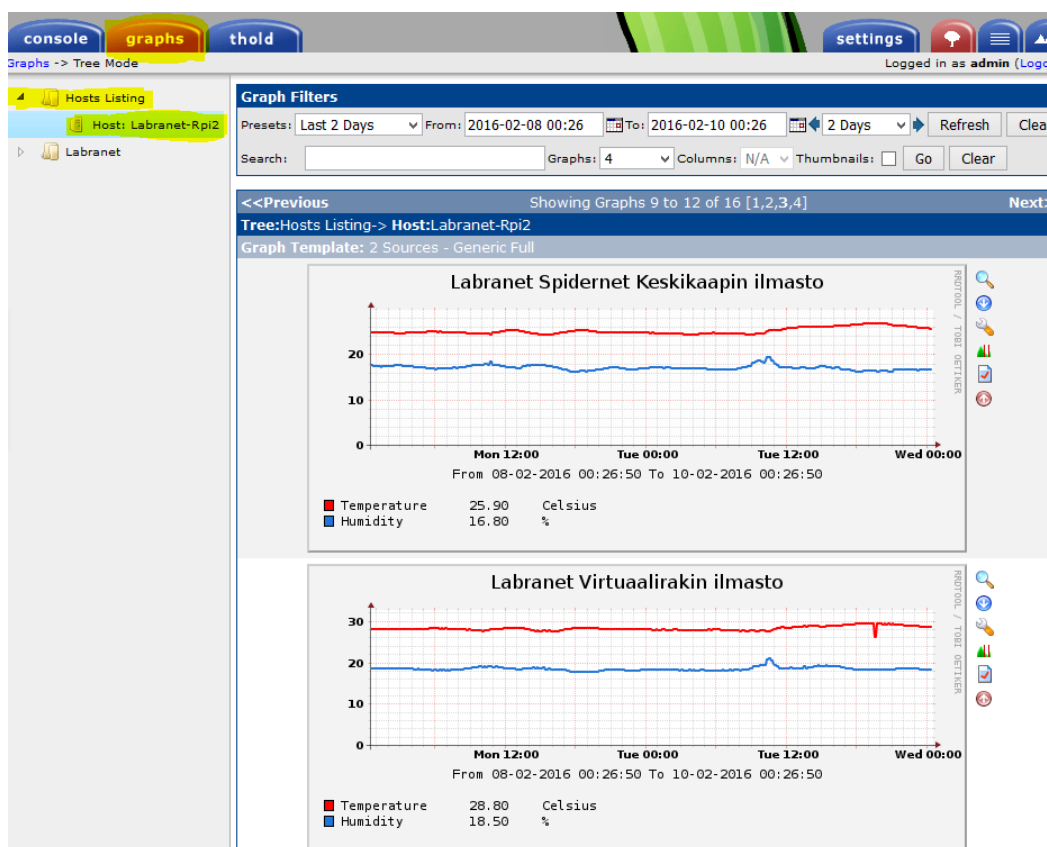
Seurattavia kohteita lisätään ”*Management*” tunnisteesta alta löytyvän ”*Data Sources*”-osion takana. Seurattavia kohteita voidaan lisätä valitsemalla *Add* ja seuraamalla ohjattua kohteen lisäystä. Kuviossa 42 näkyy seurattavan kohteen lisääminen käyttäen *SNMP - Generic OID Template* -mallia.

Kuvio 42 Seurattavan kohteen lisääminen Cactiin

6.5.3. Kuvio

Seurattavista kohteista piirretään kuvioita asettamalla kohteiden datalähteet kuviomalleihin. *Kuvioiden* tekeminen tapahtuu ”*Management*” tunnisteesta alta löytyvän ”*Graph Managementin*” takana. *Kuvioita* voidaan luoda valitsemalla *Add* ja seuraamalla ohjattua kuvion lisäämistä.

Lisäyksen jälkeen luotuja *kuvioita* voi seurata siirtymällä yläreunassa näkyvän ”*Graphs*” välilehden taakse. Kuviossa 43 on *Graphsin* näkymä ja kaksi järjestelmään syötettyä *kuvioita*. Kuvioon 43 on myös maalattu kuinka näkymään pääsee.



Kuvio 43 Monitorointinäkymä Cactissa

6.6. Näkymän muokkaaminen

Graphs osion puu-näkymä on käyttäjän muokattavissa. Muokkaaminen tapahtuu ”*Management*” tunnisteiden alta ”*Graph Trees*” osion takana. Näkymään voidaan asettaa *tunnisteita*, *laitteita*, tai *kuvioita*. Suoraan *tunnisteiden* alle asetetut *kuviot* näytetään, kun *tunniste* on aktiivisena *Graphs*-näkymässä. Jos näkymässä on aktiivisena *laite*, näytetään kaikki siltä laitteelta arvoja sisältävät *kuviot*. *Tunnisteiden* alla olevien *kuvioiden* järjestykseen voidaan myös vaikuttaa. *Kuviot* voidaan järjestellä manuaalisesti, aakkosellisesti, etumerkittömästi tai numeraalisesti.

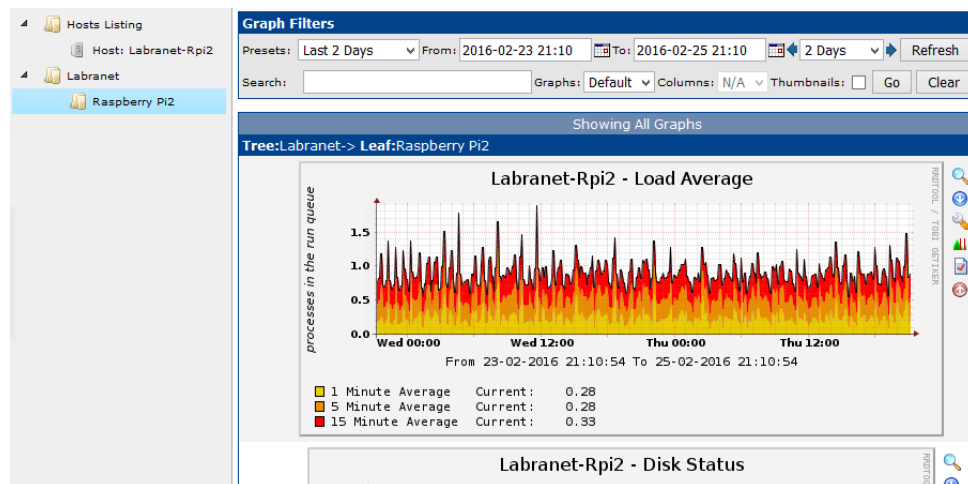
Kun *Graph Trees* valitaan ensimmäiseksi, täytyy asettaa ylin *tunniste* tai *tunnisteet*. Oletuksena ylimmäinen *tunniste* on *Default Tree*, jonka alle on asetettu Cactin isäntälaitte. Uuden *tunnisteiden* voi luoda painamalla *Add*, jolloin *tunnisteelle* annetaan nimi sekä valitaan *tunnisteiden* alle asetettavien *kuvioiden* järjestelyasetus. *Kuvioiden*

tai laitteiden lisääminen tapahtuu valitsemalla *tunniste*, jonka alle niitä halutaan lisätä. Kuviossa 44 on näkymä kun yksi *tunniste* on valittu. Jos *ylätunnisteen* alle halutaan lisätä jotain, painetaan oikeasta yläkulmasta **Add**. Jos järjestely on manuaalinen, voidaan kohteita siirrellä käyttäen näkymässä olevia nuolia. Jos ylemmän *tunnisteen* alle lisätään muita *tunnisteita*, kohteiden lisääminen näiden *tunnisteiden* alle tapahtuu painamalla *tunnisteen* nimen vieressä olevaa **Add** linkkiä.

Tree Items			Add
Expand All Collapse All			
Item	Value		
<input checked="" type="checkbox"/> Raspberry Pi2 (Add)	Heading	▼ ▲	×
Labranet-Rpi2 - Load Average	Graph	▼ ▲	×
Labranet-Rpi2 - Disk Status	Graph	▼ ▲	×
Labranet-Rpi2 - Disk Activity	Graph	▼ ▲	×
Labranet-Rpi2 - Disk I/O per second	Graph	▼ ▲	×
Labranet-Rpi2 - Disk Read/Write	Graph	▼ ▲	×
Labranet-Rpi2 - RAM Status	Graph	▼ ▲	×
Labranet-Rpi2 - Logged in Users	Graph	▼ ▲	×
Labranet-Rpi2 - Processes	Graph	▼ ▲	×
Labranet Huoneenilmasto	Graph	▼ ▲	×
Labranet Spidernet Keskikaapin ilmasto	Graph	▼ ▲	×
Labranet Virtuaalirakin ilmasto	Graph	▼ ▲	×
Labranet UPS System Status	Graph	▼ ▲	×
Labranet UPS Battery Status	Graph	▼ ▲	×
Labranet UPS System Load	Graph	▼ ▲	×
Labranet UPS Line Information	Graph	▼ ▲	×

Kuvio 44 Cactin Graph-näkymän muokkaus

Graphs näkymässä tunnisteet ja laitteet näkyvät vasemmalla puumaisena rakenteena. *Kuviot* piirretään keskelle näkymää, aikaisemmin asetetussa järjestyksessä. Kuviossa 45 on *Graphs* näkymä kuvion 44 asetuksilla.



Kuvio 45 Cactin muokattu Graph-näkymä

6.7. Tholdin käyttäminen

Hälytykset rakentuvat kahdesta osasta: tholdin asetuksista (*Settings & Device*), sekä yksittäisten kohteiden raja-arvoista (*Thresholds*).

6.7.1. Perusasetukset

Ensimmäiset asetukset löytyvät ”*Configuration*” tunnisteiden alla olevan ”*Settings*” osion takaa. *Thresholds* välilehden valitsemalla voidaan ohjata tholdin globaaleja sekä tekstiviesti hälytyksen asetuksia. Osa muokattavista asetuksista on näkyvissä kuviossa 46. Jos oletusasetuksista on otettu asetuksia pois käytöstä, niin niitä koskevat tekstiviesti hälytys asetukset eivät tule voimaan.

Cacti Settings (Thresholds)	
General	
Disable All Thresholds Checking this box will disable Alerting on all Thresholds. This can be used when it is necessary to perform maintenance on your network.	<input type="checkbox"/> Disable All Thresholds
Disable Legacy Notifications Checking this box will disable Legacy Alerting on all Thresholds. Legacy Alerting is defined as any Specific Email Alerts not associated with a Notification List.	<input type="checkbox"/> Disable Legacy Notifications
Default Status Default Threshold Filter Status	Any ▼
Base URL Cacti base URL	http://192.168.0.5/cacti/
Thresholds Per Page Number of thresholds to display per page	30 ▼
Log Threshold Breaches Enable logging of all Threshold failures to the Cacti Log	<input type="checkbox"/> Log Threshold Breaches
Log Threshold Changes Enable logging of all Threshold changes to the Cacti Log	<input type="checkbox"/> Log Threshold Changes
Debug Log Enable logging of debug messages with Thold	<input type="checkbox"/> Debug Log
Store Log for x days Keep the database threshold logs for this number of days.	31
Default Alerting Options	
Weekend exemptions If this is checked, thold will not run on weekends.	<input type="checkbox"/> Weekend exemptions
Dead Hosts Notifications Enable Dead/Recovering host notification	<input checked="" type="checkbox"/> Dead Hosts Notifications
Default Trigger Count Default number of consecutive times the Data Source must be in breach of the Threshold for an Alert to be raised	1
Re-Alerting Repeat Alert after specified number of poller cycles.	1

Kuvio 46 Tholdin globaalit asetukset

Asetukset, joilla muokataan asetettujen laitteiden saatavuuden seuraamista löytyvät *Poller* välilehden takaa, kt.s kuvion 47 ylempi osa. Nämä ovat oletusasetukset, jotka kopioidaan uudelle laitteelle luomishetkellä. Jokaisen asetetun laitteen asetuksista voidaan myös muokata, kuinka saatavuus tarkistetaan ja kuinka thold ilmoittaa saatavuuden muutoksista, kts. kuvion 47 alempi osa.

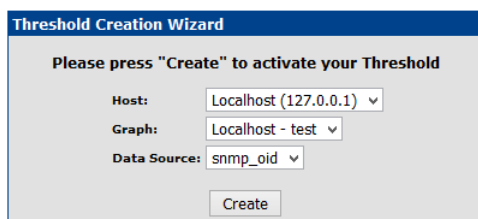
Host Availability Settings	
Downed Host Detection The method Cacti will use to determine if a host is available for polling. <i>NOTE: It is recommended that, at a minimum, SNMP always be selected.</i>	SNMP Uptime ▼
Ping Type The type of ping packet to sent. <i>NOTE: ICMP requires that the Cacti Service ID have root privileges in Unix.</i>	UDP Ping ▼
Ping Port When choosing either TCP or UDP Ping, which port should be checked for availability of the host prior to polling.	23
Ping Timeout Value The timeout value to use for host ICMP and UDP pinging. This host SNMP timeout value applies for SNMP pings.	400
Ping Retry Count The number of times Cacti will attempt to ping a host before failing.	1
Host Up/Down Settings	
Failure Count The number of polling intervals a host must be down before logging an error and reporting host as down.	1
Recovery Count The number of polling intervals a host must remain up before returning host to an up status and issuing a notice.	1

Device [edit: Localhost]	
General Host Options	
Description Give this host a meaningful description.	Localhost
Hostname Fully qualified hostname or IP address for this device.	127.0.0.1
Host Template Choose the Host Template to use to define the default Graph Templates and Data Queries associated with this Host.	Local Linux Machine ▼
Number of Collection Threads The number of concurrent threads to use for polling this device. This applies to the Spine poller only.	1 Thread (default) ▼
Disable Host Check this box to disable all checks for this host.	<input type="checkbox"/> Disable Host
Thold Up/Down Email Notification Which Notification List(s) of should be notified about Host Up/Down events?	Global List ▼
Availability/Reachability Options	
Downed Device Detection The method Cacti will use to determine if a host is available for polling. <i>NOTE: It is recommended that, at a minimum, SNMP always be selected.</i>	SNMP Uptime ▼
Ping Timeout Value The timeout value to use for host ICMP and UDP pinging. This host SNMP timeout value applies for SNMP pings.	400
Ping Retry Count After an initial failure, the number of ping retries Cacti will attempt before failing.	1

Kuvio 47 Laitteen saatavuuden seuranta asetukset Cactissa

6.7.2. Raja-arvon luominen

Raja-arvo asetusten luominen tapahtuu ”*Management*” tunnisteiden alla olevan ”*Thresholds*” osion takaa. Luominen tapahtuu painamalla **add** oikeasta yläkulmasta ja seuraamalla ohjattua raja-arvon luomista. Ensimmäisenä valitaan *kuvio*, jonka arvoja halutaan seurata, kts. kuvio 48.



Threshold Creation Wizard

Please press "Create" to activate your Threshold

Host: Localhost (127.0.0.1) ▼

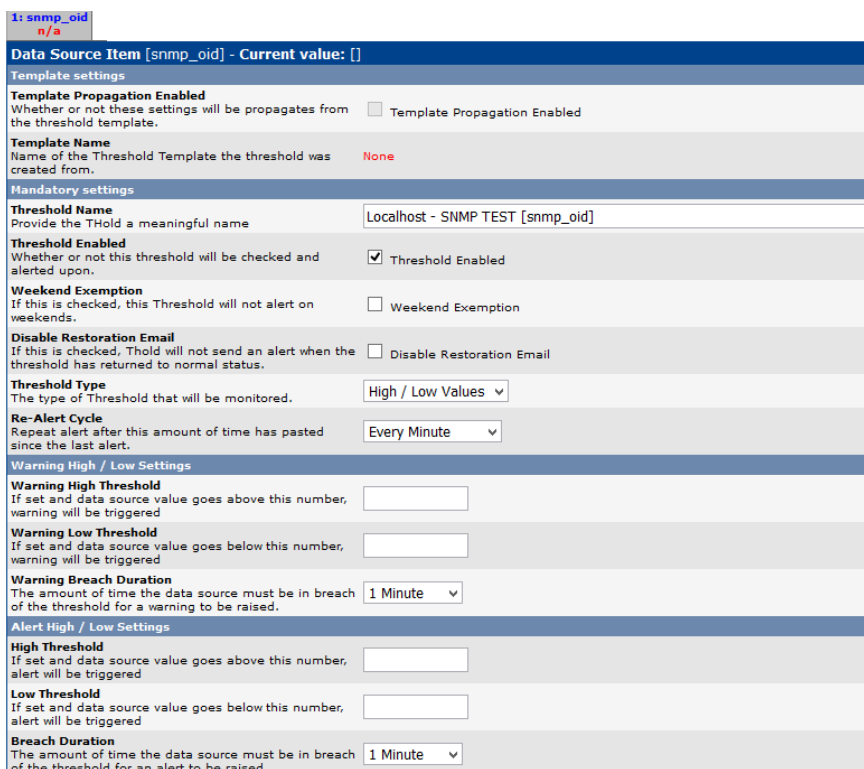
Graph: Localhost - test ▼

Data Source: snmp_oid ▼

Create

Kuvio 48 Raja-arvojen luominen Tholdiin

Seuraavassa ikkunassa voidaan asettaa raja-arvoja kaikille valitussa *kuviossa* oleville kohde arvoille. Jokaisella *kuvion* seuraamalla arvolla on oma välilehti yläreunassa, jossa eri kohteille voidaan antaa eri raja-arvoja. Raja-arvon eri asetukset ovat näkyvissä kuviossa 49.



1: snmp_oid
n/a

Data Source Item [snmp_oid] - Current value: []

Template settings

Template Propagation Enabled
Whether or not these settings will be propagated from the threshold template. ☐ Template Propagation Enabled

Template Name
Name of the Threshold Template the threshold was created from. None

Mandatory settings

Threshold Name
Provide the THold a meaningful name. Localhost - SNMP TEST [snmp_oid]

Threshold Enabled
Whether or not this threshold will be checked and alerted upon. ☒ Threshold Enabled

Weekend Exemption
If this is checked, this Threshold will not alert on weekends. ☐ Weekend Exemption

Disable Restoration Email
If this is checked, Thold will not send an alert when the threshold has returned to normal status. ☐ Disable Restoration Email

Threshold Type
The type of Threshold that will be monitored. High / Low Values ▼

Re-Alert Cycle
Repeat alert after this amount of time has passed since the last alert. Every Minute ▼

Warning High / Low Settings

Warning High Threshold
If set and data source value goes above this number, warning will be triggered.

Warning Low Threshold
If set and data source value goes below this number, warning will be triggered.

Warning Breach Duration
The amount of time the data source must be in breach of the threshold for a warning to be raised. 1 Minute ▼

Alert High / Low Settings

High Threshold
If set and data source value goes above this number, alert will be triggered.

Low Threshold
If set and data source value goes below this number, alert will be triggered.

Breach Duration
The amount of time the data source must be in breach of the threshold for an alert to be raised. 1 Minute ▼

Kuvio 49 Raja-arvon asetukset

6.7.3. Tholdin näkymä

Cactin yläreunassa on *thold* välilehti, jonka takaa löytyy tholdin käyttöliittymä. Käyttöliittymällä voidaan seurata asetettuja raja-arvoja, tholdin lokitietoja tai Cactille asetettujen laitteiden saatavuutta. Kuvion 50 ylemmässä osassa on tholdin näkymä, kun katsotaan raja-arvoja. Asetetun raja-arvon edessä olevat linkki ikonit ovat vasemmalta oikealle: *Edit Threshold*, *Disable/Enable Threshold*, *View Graph*, *View Threshold History*. Kuvion 50 alemmassa osassa laitteiden saatavuus näkymä, kun Cactille on asetettu yksi laite seurattavaksi.

Thresholds





Log

Host Status

Threshold Status

Template: AllHost: AllStatus: AllRows: 30Search:GoClear

<< PreviousShowing Rows 1 to 1 of 1 [1]Next >>

Actions	Name	ID	Type	Trigger	Duration	Repeat	Warn Hi/Lo	Alert Hi/Lo	BL Hi/Lo	Current	Triggered**	Enabled
   	Localhost - SNMP TEST [snmp_oid]	1	High/Low	1 Minute	N/A	Every Minute	-/-	100/0	N/A	-	no	Enabled

<< PreviousShowing Rows 1 to 1 of 1 [1]Next >>

Alarm	Baseline Alarm	Warning	Notice	Ok	Disabled
-------	----------------	---------	--------	----	----------

Thresholds



Log

Host Status

Device Status

Type: AllStatus: AllRows: 30Search:GoClear

<< PreviousShowing Rows 1 to 1 of 1 [1]Next >>

Actions	Description**	ID	Graphs	Data Sources	Status	Event Count	Hostname	Current (ms)	Average (ms)	Availability
 	Localhost	1	1	4	Up	0	127.0.0.1	16.16	16.54	50.94

<< PreviousShowing Rows 1 to 1 of 1 [1]Next >>

Down	Up	Recovering	Unknown	Not Monitored	Disabled
------	----	------------	---------	---------------	----------

Kuvio 50 Tholdin käyttöliittymä

7. Pohdinta

7.1. Tavoitteet ja tulokset

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa yksinkertainen järjestelmä, jolla laitetilän ilmaston tilaa voidaan seurata SNMP:llä sekä graafisella näkymällä. Työn aikana toimeksiantaja halusi myös mahdollisuuden luoda raja-arvoilla laukeavia tekstiviesti hälytyksiä.

Vahvuus Linux-käyttöjärjestelmien tavassa ajaa ohjelmia oli hyvin tärkeä työn onnistumisen kannalta. Linuxissa ohjelmille on määritetty syöttö- ja tulostusvirtoja, joilla ohjelmat kommunikoivat muun ympäristön kanssa. Eri ohjelmien virtoja voidaan asettaa jonoon tai putkeen, luoden uusia toimintoja. Bash-skripteillä onnistuttiin jo olemassa olevia ohjelmia hyväksi käyttäen luomaan hyvin monipuolisia toimintoja, joilla seurattavat kohteet saatiin kerättyä ja ulos käyttäjälle.

Toimeksiantajan laitetilän seuranta toteutettiin luomalla joukko skriptejä, joilla ilmaston tila on noudettavissa SNMP:tä käyttäen. Skriptejä luotiin myös lukemaan laitetilassa olevan keskeytymättömän virransyötön tilatietoja. Skriptien kirjoittaminen oli hyvin mielenkiintoinen kokemus, ennen työtä ohjelmointi ja skriptien luominen oli rajoittunut hyvin yksinkertaisten ohjelmien käynnistämisen automatisointiin tai vähäiseen muokkaamiseen.

Työn aikana tapahtuneen muistikortin korruptoitumisen jälkeen etäkyselyyn lisättiin myös mahdollisuus seurata yksinkertaisia muistikortin aktiivisuus tietoja. Skriptien ohjaaminen ja uudelleen asentaminen tehtiin helpoksi, joten järjestelmän laajentaminen ja uudelleenkäyttö on yksinkertaista. Toimeksiantajalle luovutetun järjestelmän etäkysely skriptien asetukset nähtävissä kuviossa 51.

```
GNU nano 2.2.6      File: /usr/local/rpi-ilmasto/bin/rpi.cfg

oid
2302 4 1
2302 17 2
2302 14 3
apc on
disk on
```

Kuvio 51 Toimeksiantajan rpi-ilmasto asetukset

Simplen, edellisistä asetuksista, luoma OID alipuu kuljettuna kuviossa 52.

```
/usr/local/rpi-ilmasto/log $ sudo snmpwalk
calhost
.0 = STRING: "rpi-ilmasto system"
.1 = STRING: "dht-ilmasto sensors"
.1.0 = STRING: "1"
.1.1 = STRING: "18.0"
.1.2 = STRING: "16.6"
.1.3 = STRING: "28.1"
.1.4 = STRING: "10.7"
.1.5 = STRING: "23.4"
.1.6 = STRING: "8.8"
.2 = STRING: "ds18-ilmasto sensors"
.2.0 = STRING: "2"
.3 = STRING: "apc-ilmasto values"
.3.0 = STRING: "0"
.3.1 = STRING: "0"
.3.2 = STRING: "29.1"
.3.3 = STRING: "3"
.3.4 = STRING: "13.0"
.3.5 = STRING: "13.5"
.3.6 = STRING: "12.0"
.3.7 = STRING: "100.0"
.3.8 = STRING: "5"
.3.9 = STRING: "232.0"
.3.10 = STRING: "230"
.3.11 = STRING: "180.0"
.3.12 = STRING: "266.0"
.4 = STRING: "system disk stats"
.4.0 = STRING: "3.63"
.4.1 = STRING: "0.00"
.4.2 = STRING: "23.07"
.4.3 = STRING: "0"
.4.4 = STRING: "1384"
.4.4 = No more variables left in this MIB View (It is past the end of th
e MIB tree)
```

Kuvio 52 Toimeksiantajalle asennetun rpi-ilmaston alipuu kuljettuna

Skriptit myös kommentoitiin helpottamaan niiden ymmärtämistä ja mahdollista laajentamista tai uudelleenkäyttöä tulevaisuudessa.

SNMP-standardeihin tutustuminen aiheutti jonkin verran ihmettelyä. SNMP on, viestintäprotokollaa lukuun ottamatta, nimensä vastaisesti hyvin monimutkainen standardien perhe. SNMP määrittelee viestinnän lisäksi niin tietokantoja kuin tietoturva asetuksia. Toisekseen, vaikka SNMP:tä käytetään hyvin monessa paikassa,

protokollan implementointi on usein tehty hyvin laiskasti, joka oli nähtävissä monissa dokumentoinneissa. Toisaalta, jos verkonhallinnan halutaan tukea enemmän kuin hyvin yksinkertaisia tilatietojen seuranta, SNMP:lle on ominaisuuksiltaan laajempia ja uudempia vaihtoehtoja.

Hyvin pienen kaistan käyttönsä vuoksi SNMP:llä voisi olla tulevaisuudessa käyttöä esineiden internet tehtävissä. IoT viestintä tapahtuu usein langattomasti tai laitteet ovat hyvin heikkotehoisia, joten yhteydet voivat olla epäluotettavia ja laskentatehon käyttö rajattua. SNMPv3 tukee myös yhdeyden salaamista, mitä kaikki muut viestintä protokollat eivät tue.

Ilmaston muutoksien seuranta mahdollistettiin RRDTool-ohjelman käyttöliittymänä toimivalla Cacti-valvontaohjelmistolla. RRDTool itse on hyvin mielenkiintoinen, ja hyvin onnistuneesti suunniteltu ja rakennettu ohjelma. RRDTool oli omiaan juurikin ilmaston arvojen tallentamisessa pitkiltä aikaväleiltä sekä kerättyjen arvojen käsittelemisessä ja esittämisessä. Cacti itse ei ole ominaisuuksiltaan kovin kummoinen ohjelmisto, mutta on erinomainen rajatuissa tehtävissä joissa täytyy tallentaa numeraalisia arvoja pitkiltä aikaväleiltä ja piirtää näiden muutoksista kuvioita. Tämän lisäksi Cactin toimintaa voi myös laajentaa, kuten työssä tehtiin.

Oletusasetuksilla Cacti seuraa kohteita 5, 10 tai 15 minuutin tarkkuudella. Asetuksia muokattiin, jotta seurannan tarkkuus saatiin yhteen minuuttiin. Cactiin luotiin lisäksi joukko kuvio malleja, joilla Cactin laajempi käyttäminen helpottuu. Cacti seuraa luovutushetkellä 34 eri kohdetta, joista ylläpidetään 15 eri kuviota. Työn aikana toimeksiantajalle tuli ajatus käyttää ylimääräistä 3G modeemia tekstiviesti hälytysten lähettämiseen. 3G modeemin käyttäminen Raspberry Pi:ssä on hyvin yleistä, joten sen käyttöönotto oli helppoa. Cactin laajentaminen hälytysjärjestelmällä onnistui myös melko kivuttomasti. Linuxin käyttäjäoikeuksien hallinta aiheutti hieman kompastelua. Hälytysjärjestelmä laajennus Cactille löytyikin helposti ja tämän laajennuksen toimintaan lisättiin tekstiviesti hälytys.

7.2. Kehitysideoita

Selkein kehitys tai laajennus kohde on *simplen* ylläpitämän alipuun laajentaminen. Simple-skriptiin on kommentoitu pieni ohje kuinka sen ylläpitämää alipuuta voidaan laajentaa. Työssä on tällä hetkellä tuki APC:n UPS-laitteille, mutta esimerkiksi laitetilan laitteiden sähkönkulutusta ei seurata.

Anturit on tällä hetkellä yhdistetty järjestelmään langallisella yhteydellä. Pienlaitteille on kehitetty erilaisia langattomia protokollia ja laitteita (esim. 6LoWPAN, Zigbee tai Xbee), joilla anturit voidaan asettaa laitteiden saataville langattomana. Järjestelmää voisi laajentaa käyttämällä langattomia protokollia/laitteita, jolloin antureita saadaan asennettua helpommin ja laajemmalle alueelle. Rasbianin kerneli tukee oletuksena ainoastaan yhtä 1-Wire GPIO-pinniä, mutta usean pinnin käyttöönotto pitäisi olla mahdollista tekemällä muutoksia kerneliin.

Cactille luotu tekstiviesti hälytys tukee ainoastaan globaalia asetusta. Laite tai kuvio kohtaiset hälytysasetukset ja puhelinnumerot voisivat olla hyödyllinen työkalu, jos järjestelmää laajennetaan. Cactin tekstiviesti ilmoitus tukee myös ainoastaan hälytyksiä (alert), eikä varoituksia (warning). Varoituksen lisääminen omalla puhelinnumero listalla mahdollistaisi eritasoiset ilmoitukset järjestelmältä käyttäjille. Esimerkiksi varoitukset voitaisiin lähettää alemman tason työntekijöille, kun hälytyksen lähetetäisiin korkeamman tason työntekijöille.

Lähteet

1-Wire Tutorial. N.d. Maxim Integrated:n 1-Wire opetusvideo. Viitattu 31.1.2016
<https://www.maximintegrated.com/en/products/1-Wire/flash/overview/>

Adam Kropelin & Kern Sibbald. 2015. APCUPSD User Manual. Viitattu 24.2.2016.
<http://www.apcupsd.org/manual/manual.html>

AM2302 Product Manual. N.d. Aosong Electronics Co.,Ltd:n tuote manuaali. Viitattu 31.1.2016
<http://akizukidenshi.com/download/ds/aosong/AM2302.pdf>

AVR318: Dallas 1-Wire master. 2004. Atmel Corporationin 1-Wire isäntälaitteen ohje. Viitattu 31.1.2016
<http://www.atmel.com/images/doc2579.pdf>

Berry, Roman, Adams, Pasnak, Conner, Scheck & Braun. 2013. The Cacti Manual Viitattu 2.2.2016
<http://www.cacti.net/downloads/docs/html/>

Bruey D. 2005. SNMP: Simple? Network Management Protocol. Viitattu 23.2.2016.
http://www.rane.com/pdf/ranenotes/SNMP_Simple_Network_Management_Protocol.pdf

Dinangkur Kundu ja S. M. Ibrahim Lavlu. 2009. Cacti 0.8 Network Monitoring: Monitor Your Network with Ease!. Packt Publishing. Viitattu 2.2.2016 Books24x7.

Downloads. N.d. Raspberry Pi Foundation:n verkkosivut. Viitattu 2.2.2016
<https://www.raspberrypi.org/downloads/>

DS18B20 Programmable Resolution 1-Wire Digital Thermometer. 2008. Maxim Integrated:n tuotteen datalehti. Viitattu 31.1.2016
<https://www.adafruit.com/datasheets/DS18B20.pdf>

Gammu integration with Cacti Threshold. 2102. Nimimerkki HendryLeo. Viitattu 10.2.2016.
<http://anythingopensource.blogspot.fi/2012/02/gammu-integration-with-cacti-threshold.html>

Godard S. N.d. Systat ohjelmisto paketin dokumentaatio sivusto. Viitattu 24.2.2016.
<http://sebastien.godard.pagesperso-orange.fr/documentation.html>

GPIO. N.d. Raspberry Pi Foundation:n verkkosivut. Viitattu 25.1.2016
<https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/README.md>

Hoda M. 2005. Cisco Network Security Troubleshooting Handbook. Viitattu 25.2.2006.
http://ciscodocuments.blogspot.fi/2011/05/chapter-05-implementing-high_20.html

Installing the Plugin Architecture. N.d. Cacti dokumentaatio sivusto. Viitattu 19.2.2016 http://docs.cacti.net/manual:087:1_installation.9_pia

Introduction to SNMP. 3.4.2015. Asante yrityksen SNMP manuaali. Viitattu 15.2.2016. <http://www.asante.com/downloads/productdocuments/snmp.pdf>

Kaario K. 2002. TCP/IP -verkot, 270-287. Porvoo: WS Bookwell. Viitattu 10.2.2016

Liu T. N.d. Aosong Electronics Co.,Ltd:n tuote manuaali. Viitattu 31.1.2016 <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

Making of Pi. N.d. Raspberry Pi Foundation:n verkkosivut. Viitattu 25.1.2016 <https://www.raspberrypi.org/about/>

NOOBS. N.d. Raspberry Pi Foundation:n verkkosivut. Viitattu 2.2.2016 <https://www.raspberrypi.org/documentation/installation/noobs.md>

Oetiker T. 2015. RRDTool Documentation. Viitattu 15.2.2016. <http://oss.oetiker.ch/rrdtool/doc/rrdtool.en.html>

OID Repository. N.d. OID kohteiden tietokanta sivusto. Viitattu 10.2.2016. <http://www.oid-info.com/#oid>

OS Market Share. 1/2016. W3Cook verkkosivut. Viitattu 7.1.2016 <http://www.w3cook.com/os/summary/>

Overview of 1-Wire Technology and its use. 19.6.2008. Maxim Integrated:n verkkosivut. Viitattu 31.1.2016 <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1796>

PHP Manual. N.d. PHP kielen kehittäjien verkkosivut. Viitattu 8.2.2016 <http://php.net/manual/en/index.php>

Rasbian. N.d. Raspberry Pi Foundation:n verkkosivut. Viitattu 2.2.2016 <https://www.raspberrypi.org/documentation/rasbian/>

RFC 2578. 04/1999. Structure of Management Information Version 2 (SMIPv2). Viitattu 23.2.2016. <https://tools.ietf.org/html/rfc2578>

RPi Hardware History. N.d. Sulautettujen Linux järjestelmien yhteisö wiki. Viitattu 25.1.2016 http://elinux.org/RPi_HardwareHistory

RPi Hardware. N.d. Sulautettujen Linux järjestelmien yhteisö wiki. Viitattu 25.1.2016 http://elinux.org/RPi_Hardware

RPi Low-level peripherals. N.d. Sulautettujen Linux järjestelmien yhteisö wiki. Viitattu 25.1.2016 http://elinux.org/RPi_Low-level_peripherals

Screen Shots. N.d. Cactin verkkosivut. Viitattu 2.2.2016
<http://www.cacti.net/screenshots.php>

Shield D. 2011. Net-SNMP Frequently Asked Questions. Viitattu 29.2.2016
<http://www.net-snmp.org/docs/FAQ.html>

SNMPD.CONF 2002. snmpd.conf tiedoston manpage. Viitattu 1.3.2016
<http://www.net-snmp.org/docs/man/snmpd.conf.html>

StatCounter Global Stats. 1/2016. StatCounterin verkkosivut. Viitattu 7.2.2016
<http://gs.statcounter.com>

The GNU Project. 1.12.2015. GNU-käyttöjärjestelmän verkkosivut. Viitattu 7.1.2016
<https://www.gnu.org/gnu/thegnuproject.html>

TM 5-693. 2002. Yhdysvaltojen sotavoimien UPS -järjestelmän valinta ja asennus manuaali. Viitattu 25.2.2016.
https://www.wbdg.org/ccb/ARMYCOE/COETM/tm_5_693.pdf

What is Apache?. N.d. Apachen dokumentaatio sivusto. Viitattu 8.2.2016
http://wiki.apache.org/httpd/FAQ#What_is_Apache.3F

What is Cacti?. N.d. Cactin verkkosivut. Viitattu 2.2.2016
http://www.cacti.net/what_is_cacti.php

What is Linux. N.d. Linux Foundationin verkkosivut. Viitattu 7.2.2016
<http://www.linuxfoundation.org/what-is-linux>

What is Raspberry Pi?. N.d. Raspberry Pi Foundation:n verkkosivut. Viitattu 25.1.2016
<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

Who Writes Linux. 2015. Linux Foundationin verkkosivut. Viitattu 7.2.2016
<http://www.linuxfoundation.org/publications/linux-foundation/who-writes-linux-2015>

LIITTEET

Liite 1 rpi-ilmasto manpage

rpi-ilmasto(1) Raspberry Pi SNMP Ilmastonseuranta rpi-ilmasto(1)

NAME

rpi-ilmasto - Raspberry Pi SNMP Ilmastonseuranta

SYNOPSIS

rpi-ilmasto [start | stop | status | edit | help]

DESCRIPTION

rpi-ilmasto has scripts that queries DHT-11, DHT-22, AM2302, DS18B20 sensors or APC manufactured UPS and makes them available by SNMP protocol. The software creates SNMP tree for user to query, the tree is built from root OID chosen by the user and two digit OID x.y that selects the sensors or UPS. X selects the sensor type or UPS (DHTs or 1-Wire DS18B20s) and Y selects a specific sensor by number chosen by user.

dht-ilmasto and ds18-ilmasto read the configuration file, from which they select which sensors to read and what number to use to identify them. When sensors give a value to the scripts, they are placed to a data base. apc-ilmasto reads the configuration file to see if the APC UPS needs to be queried, if [apc on] cannot be found in the configuration it does not run. iostat-ilmasto reads the configuration file to see if basic disk activities need to be watched, if [disk on] is not found in the configuration the script will not run. null-ilmasto is recommended to be run by crontab to ensure, in the event that sensors stop responding, that current value goes to null. It resets daily errors as well. simple is the script that answers to SNMP get queries if they are passed to it.

Options

start Starts the scripts that query the sensors or APC UPS.

stop Stops the scripts that query the sensors or APC UPS.

status Checks if any of the scripts are currently running.

edit Opens the configuration file for the scripts.

help Opens this manual page

CONFIGURATION

Options

oid OID

Used to choose the root oid for the SNMP protocol to query and respond to. OID must begin with a period [.] and end with a number.

MODEL GPIO NUMBER

This setting is for DHT/AM type temperature/humidity sensors, MODEL is 11, 22 or 2302 [for the models DHT-11, DHT-22 or AM2302]. GPIO is the General Purpose I/O to which the sensors DATA pin has been connected to. NUMBER is what you wish to define as the identification for the sensor. It also acts as the last digit of the OID that can be used to query this sensor with SNMP. DHT sensors have OID .1.n. To find the right value: Temperature is (NUMBER * 2 - 1) and humidity is (NUMBER * 2).

ds18 HWID NUMBER

This setting is for 1-Wire sensors, as the expected sensor is DS18B20, it is set as ds18. HWID is the hardware id of the sensor for the script to identify it and NUMBER is what you wish to define as the identification for the sensor. It also acts as the last digit of OID that can be used to query this sensor with SNMP. 1-Wire sensors have OID .2.n.

apc on

This is a setting that allows apc-ilmasto to run. If setting is enabled Simple creates .3.x OID branch from 0 to 12, where 0 is daily errorcount and the rest are in the following order, starting with 1: STATUS, TIMELEFT, MINTIMEL, LOADPCT, BATTV, NOBATTV, BCHARGE, MBATTCHG, LINEV, NOMINV, LOTRANS, HITRANS.

disk on

This is a setting that allows iostat-ilmasto to run. If setting is enabled, Simple will create .4.x OID branch from 0 to 4, where 1 is IOPS, 2 is Read speed kB/s, 3 is Write speed kB/s, 4 Bytes read, 5 Bytes written.

EXAMPLE

Root OID is selected as .1.2.3 and there are three sensors, DHT-22, AM2302 and DS18B20. DHT type sensors are in GPIOs 4 and 17. DS18B20 sensor has HWID 10-10101010, these sensors are given NUMBERS, that are smallest possible, so 1,2 and 3. DHT sensors would be found with OIDs 1.2.3.1.1-2 and 1.2.3.1.3-4. DS18B20 gets OID 1.2.3.2.1.

```
#Configuration file rpi.cfg
oid .1.2.3
22 4 1
2302 17 2
ds18 10-10101010 1
```

SETUP

Dependencies

python To query DHT sensors with Adafruits python libraries.

Adafruit Python DHT Library github.com/adafruit/Adafruit_Python_DHT

snmpd Required so snmp queries can be made

apcupsd if you wish to access information from your APC UPS.

iostat if you wish to access the basic disk activity information. (Part of sysstat package)

rrdtool Required to create the .rrd databases that are queried by SNMP

snmpd configurations

For the snmpd to pass the SNMP queries to the simple you need to edit the /etc/snmp/snmpd.conf file and add line "pass OID /bin/bash /usr/local/rpi-ilmasto/bin/simple", where OID is the root of the sensors. It must be identical to the OID in the rpi.cfg configuration file. Adding .1.x, .2.x or .3.x after it when making GET query will have simple answer with the latest value from the sensor. simple will respond if the oid tree is queried and it is possible to snmpwalk the root OID.

EXAMPLE

Following example allows you to query your chosen OID with community user rpiuser.

```
#/etc/snmp/snmpd.conf configuration example
agentAddress 161
rocommunity rpiuser
pass OID /bin/bash /usr/local/rpi-ilmasto/bin/simple
```

crontab configurations

It is recommended to run null-ilmasto with argument "00" once a day, it resets error counter to "0", recommend resetting it at midnight, giving you daily error count. Argument "rest" will set all the other databases values to unknown, it is recommended to run it few times during the day to make sure if sensors stop working that values given by SNMP go to NULL, instead of giving possibly incorrect values.

EXAMPLE

```
#crontab -e configuration example
0 0 * * * /bin/bash /usr/local/rpi-ilmasto/bin/null-ilmasto 00
59 * * * * /bin/bash /usr/local/rpi-ilmasto/bin/null-ilmasto rest
```

logrotate configurations

The scripts do not have any checks for log filesizes. It is recommended to set up a logrotate so the logfiles don't grow to absurd sizes over long period of time.

EXAMPLE

```
#/etc/logrotate.conf example
/usr/local/rpi-ilmasto/log* {
    monthly
    minsize 1M
    maxsize 5M
    missingok
    nocreate
    rotate 1
    notifempty
}
```

File locations

For the software to run correctly all files must be located in /usr/local/rpi-ilmasto/, with scripts and configuration files being in /bin, .rrd databases in /databases and log files in /log.

FILESrpi-ilmasto

Initialization script that controls the data gathering scripts or opens the configuration file for editing.

dht-ilmasto

Script that queries DHT-11, DHT-22 and AM2302 sensors and upkeeps their databases.

ds18-ilmasto

Script that queries DS18B20 sensors and upkeeps their databases.

apc-ilmasto

Script that queries UPS device and upkeeps databases from parts of the output.

iostat-ilmasto

Script that watches disk activity and places the averages of the last minute into database.

null-ilmasto

Script that resets the databases to 0 or unknown value. Giving it argument "00" will reset errorcount, "rest" will reset all other databases.

rpi.cfg

Configuration file that scripts read to find sensors to query.

simple

Script that answers to SNMP get queries with latest values from the database.

ilmasto.pid

Temporary file that is created when starting the scripts, if file does not exist scripts stop.

AdafruitDHT.py

Python code that calls DHT type sensors, it's being called from inside the dht-ilmasto script. Its a part of Adafruits DHT library and has been modified to not give text on its query strings.

Logfiles

dht.log/ds18.log/apc.log/io.log are used to log activities, errors or problems met by the scripts.

EXTENDING THE SOFTWARE**First words**

If extending the scripts is something you would like to do here are couple ideas to start with. simple and apc-ilmasto are easily extendable by anyone who has any understanding of scripting. The scripts have small amounts of comments for help in them, if you wish to add more features to the software.

More information from APC UPS.

Adding a number after 'apc on' in the rpi.cfg will extend the simple oid numbers by increasing the max value of the last OID if ROOT.3.X is queried, default and minimum value is 12. After that, simply add the wanted informations you wish to get by extending the apc-ilmasto script.

Extending the SNMP oid

Open the simple script and go to 'while \$oid1 loop' at the later half of the script. Adding additional 'cases' inside the loop can be used to add your own things for the SNMP to query. As long as the script returns three echoes SNMP will get an answer. first echo is the target OID or the next OID after target (get or getnext), second echo is information type (integrated/string), third is the value of the target.

BUGS & ISSUES

- Incorrectly set configurations can cause total failure or slow down the scripts.
- Scripts must be run as root, which can be a bit of a security issue if multiple users have access to the raspberry pi.
- Log files can grow to take quite an enourmous amount of space, use logrotate.

Version 1.1

1 February 2016

rpi-ilmasto(1)

Liite 2 Rpi-ilmaston MIB-moduuli

```

-- THIS IS EXPERIMENTAL MIB FILE
-- DO NOT SHARE WITH UNREGISTERED OID

RPIILMASTO-MIB DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
        FROM SNMPv2-CONF;

rpi-ilmasto MODULE-IDENTITY
    LAST-UPDATED "201601030000Z"
    ORGANIZATION "J.H"
    CONTACT-INFO
        "EMail: none
         phone: none"
    DESCRIPTION
        "The MIB module for rpi-ilmasto.
         Offers readable names for sensors with
         short descriptions
         DO NOT SHARE"
    REVISION   "201601030000Z"
    DESCRIPTION
        "This is the initial revision of the
         rpi-ilmasto MIB"
    ::= { xy-oid-xy }

rpi-ilmasto-null OBJECT IDENTIFIER ::= { rpi-ilmasto 0 }

-- dht-sensors

dht-sensors OBJECT IDENTIFIER ::= { rpi-ilmasto 1 }

dht-errors OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This is the current error count for dht-script"
    ::= { dht-sensors 0 }

dht-1-temperature OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This is the latest temperature value
         as given by dht sensor 1, value is in celsius"
    ::= { dht-sensors 1 }

dht-1-humidity OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This is the latest humidity value
         as given by dht sensor 1, value is indicated in percent"
    ::= { dht-sensors 2 }

dht-2-temperature OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS   read-only

```

```

STATUS    current
DESCRIPTION
    "This is the latest temperature value
    as given by dht sensor 2, value is in celsius"
::= { dht-sensors 3 }
dht-2-humidity OBJECT-TYPE
SYNTAX    String
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
    "This is the latest humidity value
    as given by dht sensor 2, value is indicated in percent"
::= { dht-sensors 4 }
dht-3-temperature OBJECT-TYPE
SYNTAX    String
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
    "This is the latest temperature value
    as given by dht sensor 3, value is in celsius"
::= { dht-sensors 5 }
dht-3-humidity OBJECT-TYPE
SYNTAX    String
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
    "This is the latest humidity value
    as given by dht sensor 3, value is indicated in percent"
::= { dht-sensors 6 }
dht-4-temperature OBJECT-TYPE
SYNTAX    String
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
    "This is the latest temperature value
    as given by dht sensor 4, value is in celsius"
::= { dht-sensors 7 }
dht-4-humidity OBJECT-TYPE
SYNTAX    String
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
    "This is the latest humidity value
    as given by dht sensor 4, value is indicated in percent"
::= { dht-sensors 8 }
dht-5-temperature OBJECT-TYPE
SYNTAX    String
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
    "This is the latest temperature value
    as given by dht sensor 5, value is in celsius"
::= { dht-sensors 9 }
dht-5-humidity OBJECT-TYPE
SYNTAX    String
MAX-ACCESS read-only
STATUS    current
DESCRIPTION
    "This is the latest humidity value
    as given by dht sensor 5, value is indicated in percent"
::= { dht-sensors 10 }

-- end of dht-sensors

```

```

-- ds18 sensors

ds18-sensors OBJECT IDENTIFIER ::= { rpi-ilmasto 2 }

ds18-errors OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the current error count for ds18-ilmasto
        script"
    ::= { ds18-sensors 0 }
ds18-1 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 1, value is in celsius"
    ::= { ds18-sensors 1 }
ds18-2 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 2, value is in celsius"
    ::= { ds18-sensors 2 }
ds18-3 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 3, value is in celsius"
    ::= { ds18-sensors 3 }
ds18-3 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 3, value is in celsius"
    ::= { ds18-sensors 3 }
ds18-4 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 4, value is in celsius"
    ::= { ds18-sensors 4 }
ds18-5 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 5, value is in celsius"
    ::= { ds18-sensors 5 }
ds18-6 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION

```

```

        "This is the latest temperature value
        as given by ds18 sensor 6, value is in celsius"
        ::= { ds18-sensors 6 }
ds18-7 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 7, value is in celsius"
        ::= { ds18-sensors 7 }
ds18-8 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 8, value is in celsius"
        ::= { ds18-sensors 8 }
ds18-9 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 9, value is in celsius"
        ::= { ds18-sensors 9 }
ds18-10 OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest temperature value
        as given by ds18 sensor 10, value is in celsius"
        ::= { ds18-sensors 10 }

-- end of ds18-sensors

-- apc-information

apc-information OBJECT IDENTIFIER ::= { rpi-ilmasto 3 }

apc-errors OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the current error count for apc-ilmasto script"
        ::= { apc-information 0 }

apc-status OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the latest status given by your UPS.
        1=ON-BATTERY, 0=ONLINE, -1=UNKNOWN"
        ::= { apc-information 1 }

apc-timeleft OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the timeleft you have in minutes if the UPS

```


switches to the battery with the current load."
 ::= { apc-information 2 }

apc-minimum-time OBJECT-TYPE
 SYNTAX String
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "This is the minimum timeleft that if reached the UPS
 will shutdown your systems to ensure your batteries
 are not damaged by undercharge"
 ::= { apc-information 3 }

apc-current-load OBJECT-TYPE
 SYNTAX String
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "This is the current load percentage of load capacity
 used, as estimated by your UPS"
 ::= { apc-information 4 }

apc-current-battery-volt OBJECT-TYPE
 SYNTAX String
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "This is the current battery voltage given
 by your UPS"
 ::= { apc-information 5 }

apc-nominal-battery-volt OBJECT-TYPE
 SYNTAX String
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "This is the nominal battery voltage as
 told by your UPS"
 ::= { apc-information 6 }

apc-current-battery-charge OBJECT-TYPE
 SYNTAX String
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "This is the current battery capacity charge percentage
 as estimated by your UPS"
 ::= { apc-information 7 }

apc-minimum-battery-charge OBJECT-TYPE
 SYNTAX String
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "This is the minimal battery charge; if reached
 the UPS will shutdown your systems, due to undercharge."
 ::= { apc-information 8 }

apc-current-input OBJECT-TYPE
 SYNTAX String
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "This is the current input voltage."
 "

```

::= { apc-information 9 }

apc-nominal-input OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the nominal input voltage as told
        by your UPS."
    ::= { apc-information 10 }

apc-minimum-input OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the minimum input voltage that if breached,
        your UPS will switch to batteries."
    ::= { apc-information 11 }

apc-maximum-input OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the maximum input voltage, that if breached
        your UPS will switch to batteries."
    ::= { apc-information 12 }

-- end of apc-information

-- disk-info

disk-iops OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the average I/O per second over last minute"
    ::= { disk-information 0 }

disk-reading OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the average reading in kB/s
        that happened over the last minute"
    ::= { disk-information 1 }

disk-writing OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the average writing in kB/s
        that happened over the last minute"
    ::= { disk-information 2 }

disk-read OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the amount of reading that was

```

```
        done over the last minute in kB"
    ::= { disk-information 3 }

disk-written OBJECT-TYPE
    SYNTAX      String
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This is the amount of writing that was
        done over the last minute in kB"
    ::= { disk-information 4 }

-- end of disk-info

END
```